The Generation of Depth Maps

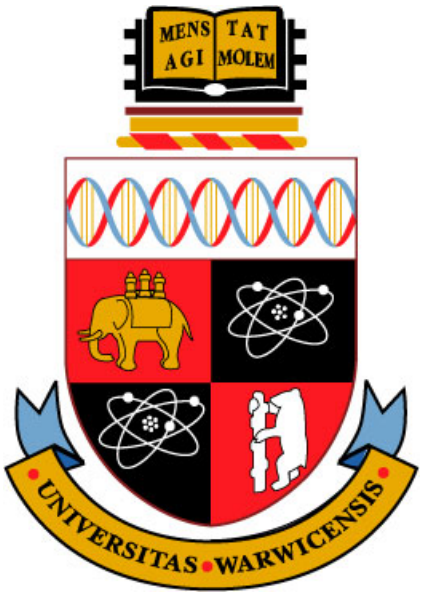
via Depth-from-Defocus

by

William Edward Crofts

A thesis submitted in partial fulfilment of the
requirements for the degree of
Doctor of Philosophy

School of Engineering
University of Warwick



September 2007

Table of Contents

Chapter 1 Introduction 1-1

1.1 Background to the relationship between depth and defocus 1-2

1.2 Structure of this thesis 1-5

1.2.1 Summary of chapter contents 1-5

Chapter 1 – References 1-9

Chapter 2 Optics and DfD Methods 2-1

2.1 Lens Systems and Defocus/Depth Relationships 2-1

2.1.1 Thin lenses 2-1

2.1.2 Defocus/Depth Relationship 2-7

2.2 Modelling Defocus 2-9

2.2.1 Point Spread Function 2-9

2.2.2 Line Spread Function 2-13

2.2.3 Edge Spread Function 2-14

2.3 Methods of using defocus to estimate depth 2-16

2.3.1 Two-image method – using different size of aperture opening 2-17

2.3.2 Two-image method – using two image detectors and telecentricity 2-19

2.3.3 Two-image method – changing focal length 2-23

2.4 Other methods of measuring depth 2-25

2.4.1 Stereo imaging 2-25

2.4.2 Photogrammetry 2-26

2.4.3 Light stripe projection 2-27

2.4.4 Laser tracker 2-28

2.4.5 Laser time-of-flight 2-29

2.4.6 Ultrasound sensors 2-30

2.4.7 Coordinate Measuring Machines (CMM) 2-30

2.5 Summary of Chapter 2 2-31

Chapter 2 – References 2-33

Chapter 3 A Summary of Influential Research in Depth-from-Defocus 3-1

3.1 Single-image with Passive Illumination Methods 3-2

3.2 Multiple-Image with Passive Illumination and Spatial Analysis Methods 3-3

3.3 Multiple-Image with Passive Illumination and Frequency Analysis Methods 3-11

3.4 Single-Image with Active Illumination Methods 3-16

3.5 Multiple-Image with Active Illumination Methods 3-18

3.6 Other Optical Methods of Estimating Depth 3-22

3.6.1 Neural Networks 3-22

3.6.2 Photogrammetry 3-23

3.7 Summary of Chapter 3 3-24

Chapter 3 – References 3-25

Chapter 4 Investigating Depth Cues in a Projected Structured Light Pattern 4-1

4.1 Initial depth cue experiments 4-2

4.2 Development of pattern analysis algorithm 4-5

4.3 Experiment 1: Depth cue variation versus depth change 4-6

4.4 Experiment 2: Investigation of depth cues in projected structured light patterns using a Foculus firewire camera 4-9

4.5 Intermediate conclusion on depth cue performance 4-12

4.6 Experiment 3: The analysis of edge profile variation versus depth change 4-12

4.6.1 Equipment for Experiment 3 4-14

4.6.2 Modified algorithm for detection and extraction of edge profile 4-14

4.6.3 Determining an edge profile parameter for evaluating depth 4-17

4.7 Summary of Chapter 4 4-24

Chapter 4 – References 4-25

Chapter 5 The Development of an Active DfD System 5-1

5.1 Initial set-up and the fundamental pattern analysis algorithm 5-2

5.1.1 Development of the algorithm to generate a 2-dimensional look-up table of standardised depth images 5-3

5.1.2 Development of an algorithm to match an edge profile with its nearest corresponding profile in the 2-D look-up table 5-7

5.2 Experiment 4: Evaluating the Depth-from-Defocus algorithm using a 2-Dimensional look-up table 5-9

5.3 Experiment 5: Evaluating the effect of variance in surface reflectance 5-12

5.4 Experiment 6: Evaluating the effect of surface angle 5-13

5.5 Experiment 7: Evaluating the effect of image averaging 5-15

5.6 The Plus U2 Data Projector – Potential problem source 5-17

5.6.1 Data Projector Operating Temperature 5-18

5.6.2 Digital Light Projection (DLP) System Technology 5-19

5.7 Variation in depth estimates across the image row 5-21

5.8 Development of a 3-D look-up table to improve the horizontal spatial accuracy of depth estimates 5-25

5.8.1 Increasing the spatial resolution of depth estimates by moving the projected structured light pattern 5-26

5.8.2 Improvement in horizontal variation of depth estimates by using the 3-D look-up table method 5-34

5.9 Experiment 8: Evaluating the effectiveness of the 3-D look-up table system 5-35

5.10 Development of a 4-D look-up table to improve the vertical spatial accuracy of depth estimates 5-37

5.11 Testing the performance of the DfD system using the 4-D look-up table algorithm on planar surfaces 5-41

5.12 Neutralising original image texture to improve depth estimate accuracy 5-43

5.13 Summary of Chapter 5 5-46

Chapter 5 – References 5-47

Chapter 6 Applications of an Active DfD System 6-1

6.1 Projecting along the optical axis 6-1

6.2 Application to 3-D shaped objects 6-4

6.2.1 Planar shapes at varied depths 6-4

6.2.2 Sloping plane 6-8

6.2.3 Sphere of white polystyrene 6-11

6.2.4 Complex shape – high-density (blue) polystyrene 6-16

6.3 Summary of Chapter 6 6-19

Chapter 7 Conclusions and Further Work 7-1

7.1 Conclusions 7-1

7.2 Summary of novel features and original contribution 7-6

7.3 Suggestions for further work 7-7

Chapter 7 – References 7-10

Appendix A MATLAB Files A-1 to A-26

Appendix B Experimental Results B-1 to B-39

Appendix C Equipment Specifications C-1 to C-4

Table of Figures

Figure 1.1 A digital image of three items at different depths……. 1-3
Figure 1.2 The effect of human vision with a fixed viewpoint…….. 1-4
Figure 2.1 Cross sections of simple spherical lenses……. 2-2
Figure 2.2 Refraction at a spherical interface……. 2-3
Figure 2.3 Refraction at a spherical interface for different source points *S*…… 2-4
Figure 2.4 Spherical Lens Geometry…….. 2-5
Figure 2.5 Imaging Geometry related to Defocus Blur……. 2-7
Figure 2.6 Camera Parameters for Defocus/Depth Relationship…….. 2-10
Figure 2.7 An Illustration of Line Spread Function…….. 2-13
Figure 2.8 An edge after blurring with a generic PSF of two widths……. 2-15
Figure 2.9 Arrangement to produce two identical images except for depth of field…….. 2-17
Figure 2.10 Spatial invariance with two image planes……. 2-20
Figure 2.11 Constant magnification using an aperture at the lens front-focal plane……. 2-21
Figure 2.12 Relationship between focus measures g_1 and g_2 and defocus measure α…….. 2-23
Figure 2.13 Geometry of two-lens system for variable focal length……. 2-24
Figure 2.14 Binocular Stereo Geometry…… 2-25
Figure 2.15 Coded photogrammetry targets fitted to a manufactured pipe…… 2-27
Figure 2.16 The Principle of Light-Stripe Projection……. 2-27
Figure 2.17 Laser Interferometer Arrangement…….. 2-28
Figure 2.18 Laser-tracker Retro-Reflective Tracking Mechanism……. 2-29
Figure 2.19 Coordinate Measuring Machine Devices and Applications……. 2-30
Figure 3.1 Simultaneous Image Capture……. 3-3
Figure 3.2 Block diagram of the Ens & Lawrence approach to DfD…….. 3-5
Figure 3.3 Schematic of Rajagopalan & Choudhuri Method…….. 3-8
Figure 3.4 From Deschenes, et al…… 3-10
Figure 3.5 Two-lens Arrangement – Subbarao…… 3-12
Figure 3.6 Estimates of depth using various SFR methods – Rajagopalan, Chaudhuri ….…..… 3-15

Figure 3.7 A DfD light source with astigmatic optics - Girod & Scherock…… 3-17
Figure 3.8 Fringe pattern projection system – Hinojosa C, et al……. 3-18
Figure 3.9 DfD system developed by Nayar et al……... 3-20
Figure 3.10 Two-image constant magnification imaging system. Nayar et al… 3-20
Figure 3.11 Shadowing by both camera and projector – Kowarschik……... 3-21
Figure 4.1 Preliminary Experimental Rig…….. 4-3
Figure 4.2 Images showing spread of projected vertical white bars due to defocus……... 4-4
Figure 4.3 Illustration of the algorithm used in 'VertStripeCalc.m' to determine parameters to be used for comparison of structured light pattern depth cues…… 4-5
Figure 4.4 Plot of Depth Cues versus Depth…….. 4-6
Figure 4.5 Plot of Pattern Intensity versus Depth……... 4-7
Figure 4.6 Plot of Mark/Space Ratio versus Depth…… 4-8
Figure 4.7 An Extract of Results from App B-2……. 4-9
Figure 4.8 Plot of Standard Deviation versus Depth (Foculus)…….. 4-10
Figure 4.9 Plot of Pattern Frequency Period versus Depth (Foculus)…… 4-11
Figure 4.10 Plot of Mark/Space Ratio versus Depth (Foculus)…….. 4-11
Figure 4.11 Variation of Edge Profile versus Depth……... 4-13
Figure 4.12 Detection and Extraction of Edge Profile…… 4-15
Figure 4.13 'Glitch' in Edge Profile …….. 4-16
Figure 4.14 Best fit cubic polynomials for defocused edge profiles…….. 4-18
Figure 4.15 Illustrating an Indirect Method of Profile to Polynomial Comparison……. 4-19
Figure 4.16 Analysis of Optimum Depth Cue Features…….. 4-20
Figure 4.17 Comparison of L1 norm residuals……... 4-21
Figure 4.18 Comparison of L2 norm residuals…… 4-22
Figure 4.19 Comparison of Sum-Difference residuals…… 4-23
Figure 5.1 Initial Camera/Projector Set-up……. 5-2
Figure 5.2 (a) Illustration of Captured Image Section
(b) Typical profile of intensity levels along a row of pixels……. 5-4
Figure 5.3 Flowchart to Illustrate the Algorithmic design of AverageCalc.m… 5-5
Figure 5.4 Flowchart of Depth Matching Algorithm…….. 5-8
Figure 5.5 Plot of Error Results from Table 5.1…….. 5-10

Figure 5.6 Plot of Standard Deviation of Depth Estimate Results from Table 5.1…… 5-11
Figure 5.7 Plot of Standard Deviation and % Error versus Background Absorption…….. 5-13
Figure 5.8 Surface Angle at 0^o……. 5-14
Figure 5.9 Surface Angle at 30^o……... 5-14
Figure 5.10 Plot of % Error in Depth Estimation versus Variation of Surface Angle……... 5-15
Figure 5.11 Plot of % Error versus No of Time-lapsed Averaged Images…….. 5-16
Figure 5.12 Detail of a Single DMD Mirror Pixel Cell…….. 5-19
Figure 5.13 The DLP System including a Digital Micromirror Device…….. 5-20
Figure 5.14 Horizontal Variation in Edge Profile Look-up Value……. 5-22
Figure 5.15 Variation in Profile Intensity Across an Image Row……... 5-23
Figure 5.16 Plot of six edge profile look-up values across image row…… 5-23
Figure 5.17 Reflection of Projected Pattern from Planar Screen…… 5-24
Figure 5.18 3-D Look-up Table…….. 5-25
Figure 5.19 Typical variation in edge profile shape as degree of edge focus changes……. 5-27
Figure 5.20 Typical row profile from planar pattern image…… 5-28
Figure 5.21 Typical row profile from defocused pattern of insufficient bar width size……... 5-28
Figure 5.22 Revised depth map for optimum pattern size…….. 5-29
Figure 5.23 Available rising edges from optimised pattern…… 5-30
Figure 5.24 Alignment of Valid Edges on Pattern-Shifted Images……. 5-32
Figure 5.25 Application of high-resolution 3-D depth map…… 5-34
Figure 5.26 Depth Estimate Values Across Image Row using 3-D Look-up Table…….. 5-36
Figure 5.27 Vertical Variation in Depth Estimation…… 5-38
Figure 5.28 Layout of 4-D Look-up Table…….. 5-39
Figure 5.29 Vertical Depth Variation using 4-D Algorithm……... 5-40
Figure 5.30 Planar Object at 90 cm……. 5-41
Figure 5.31 Performance of the 4-D DfD system on planar surfaces…….. 5-43
Figure 5.32 Comparison of 4-D DfD Performance Before & After

Texture Neutralisation……. 5-46
Figure 6.1 Projecting along optical axis…….. 6-2
Figure 6.2 Half-silvered mirror arrangement for projecting on optical axis….. 6-2
Figure 6.3 Computer-controlled motorised movement carrying object mounting/screen fixture……. 6-3
Figure 6.4 Plan view of planar object arrangements……... 6-4
Figure 6.5 Image of planar screens with non-active illumination……... 6-5
Figure 6.6 Mesh plot of depth map estimates for planar panels at varying depths…… 6-6
Figure 6.7 Extracted Areas of Depth Planes used for Analysis…….. 6-7
Figure 6.8 Plan view of sloping plane object…….. 6-8
Figure 6.9 Image of sloping plane object…… 6-9
Figure 6.10 Mesh plot of sloping plane object depth map…….. 6-9
Figure 6.11 Plot of Sloping Plane Depth Estimates…… 6-10
Figure 6.12 Depth Estimation Error for Sloping Plane Application…….. 6-11
Figure 6.13 Plan view of spherical white ball shape…….. 6-12
Figure 6.14 Image of white sphere object…….. 6-13
Figure 6.15 Mesh plot of white sphere object depth map…….. 6-13
Figure 6.16 Sample range of extracted depth map row……. 6-14
Figure 6.17 Actual depth versus estimated depth for Sphere application……. 6-15
Figure 6.18 Depth Estimate Error for Sphere Application Sample…… 6-15
Figure 6.19 Complex shape – high-density polystyrene……. 6-17
Figure 6.20 Image of complex shape under non-active illumination……. 6-17
Figure 6.21 Mesh plot of depth map estimates for complex shape…… 6-18

Table of Tables

Table 3.1 Previous work using Single Image with Passive Illumination Methods……. 3-2

Table 3.2 Previous Work Using Multiple Image with Passive Illumination and Spatial Analysis Methods……. 3-4

Table 3.3 Previous Work Using Multiple Image with Passive Illumination and Frequency Analysis Methods……... 3-11

Table 3.4 Previous Work Using Single Image with Active Illumination Methods…….. 3-16

Table 3.5 Previous Work Using Multiple Image with Active Illumination Methods……. 3-19

Table 5.1 Results of First DfD Algorithm Test (2-D LUT)…….. 5-10

Table 6.1 Depth Map Values for Varied Plane Application……. 6-7

Acknowledgments

I would like to acknowledge my sincere gratitude to my supervisor Dr Richard Staunton for the great deal of patience he has shown over the period of this part-time PhD, and for the considerable level of diligence and care he has taken in helping me to complete this thesis.

I would also like to thank Dr Christopher Claxton who generously gave me the computer-controlled, motorised movement that he had devised for the object mounting/screen fixture, and for customising the associated control software to suit my needs.

Dr James Covington provided the long-term loan of his high-resolution UEye camera and lenses. This equipment proved invaluable to the success of the research I have completed in this study, and I am extremely grateful to him for trusting me with such an expensive device over this period.

Mr Peter Kimber used rapid prototyping equipment to provide me with some test artefacts, including the complex-shaped high-density polystyrene object (featured in Chapter 6). I would like to thank him for his time and effort.

Finally, I would like to thank Mr David Robinson who gave a lot of his time and skills as a technician in the micro-machining workshop to help me construct some of the early prototypes that I worked with.

Dedication

To my wife, Sue, who has often gone without holidays, etc., to enable me to complete this difficult part-time study, but has always been entirely supportive, understanding, and helpful.

To my son, Neil, who – despite a difficult start to his working life – is showing all the signs of Crofts' determination to get there in the end, and is using enterprise, ingenuity, and character to gradually forge success for himself.

Sue and Neil have formed the centre of home, family life, and stability that has enabled me to manage the enormous amount of work required to complete this PhD in conjunction with a very busy full-time job.

Declaration

This thesis is submitted in partial fulfilment for the degree of Doctor of Philosophy under the regulations set out by the Graduate School at the University of Warwick.
This thesis is entirely composed of research completed by William E Crofts under the supervision of Dr. Richard Staunton.
None of the work presented here has been published or submitted for another degree.

Abstract

The principle aim of this study was to use the concept of image defocus related to feature depth in order to develop a system capable of converting a 2-dimensional greyscale image into a 3-dimensional depth map. An advantage of this concept (known as depth-from-defocus or simply DfD) over techniques such as stereo imaging is that there is no so-called 'correspondence problem' where the corresponding location of a feature or landmark point must be identified in each of the stereo images.

The majority – and the most successful – of previous researchers in DfD have used some variation of a 'two-image' technique in order to separate the contribution of the original scene features from the defocus effect. The best of those have achieved results typically in the range of 1% to 2% error in the accuracy of depth estimation.

This thesis presents a single-image method of generating a high-density, high-accuracy depth map via the evaluation of the edge profiles of a projected structured light pattern. A novel technique of moving the projected pattern during the image capture stage allows the development of a 4-dimensional look-up table. This technique offers a solution to one of the last remaining problems in DfD, that of spatial variance. It also uses a technique to remove the dependence of original scene reflectance. The final solution generates a depth map of up to 240,000 spatially invariant depth estimates per scene image, with an accuracy of within ± 0.5%, over a depth range of 10 cm. The depth map is generated in a processing time of approximately 14 seconds once the images are loaded.

Chapter 1

Introduction

The primary objective of the work described in this thesis is to examine methods of quantifying the defocus gradient in images, and to determine a suitable method of using the defocus effect to generate depth maps of an object in the scene. This is generally referred to as 'Depth from Defocus' or simply 'DfD'. A large range of DFD techniques have been developed (see chapter 3), but they all essentially use some means of analysing levels of defocus in the image scene in order to equate this to depth (i.e. distance between object and camera lens) and to produce a depth map of the image. In this sense they are converting a 2-D representation of the object into a 3-D representation.

The initial motivation for pursuing this technique was the author's association with a manufacturer of coordinate measuring machines (ITP, Rugby). These machines use a stylus-type contact probe that is touch-sensitive. However, customers of the company often had the need to measure artefacts that were of either soft material, or were flexible in nature (i.e. shaped fuel pipes in aircraft wings, etc), and did not suit being physically probed by a machine stylus. Other types of non-contact probe (e.g. laser-based) are available, but these still require coordinate points to be measured individually, which may be very slow compared to an image processing method that can analyse all image points in one operation. Hence, DfD was initially selected as a possible non-contact alternative. However, association with the company was lost during the early stages of this study, and it was decided to continue the investigation

of DfD methods from a more general theoretical point of view. It is now considered that the techniques described in this thesis could be developed into a number of possible applications.

1.1 Background to the relationship between depth and defocus

In one sense we all use the relationship between image defocus and depth during every waking moment of our lives. However – as with many other human physiological systems – it takes place entirely automatically and we are not really conscious of it.

The human vision system actually generates badly degraded images via the retina of the eye. However, it is apparent that biological vision systems have developed in this way for a purpose [1], [2], [3]. Whilst it would have been easier for nature to develop eyes that were much closer to a perfect lens, it seems that we use the fact that – within our field of view – the only area generally in focus is the immediate area on which we are concentrating (see Figures 1.1 and 1.2). Features in the area outside of this narrow field of focus are defocused to a degree that is dependent upon their depth (i.e. distance from the viewer). It is apparent that our brain learns to estimate depth via two sources. One is from the stereo effect provided by the pair of human eyes when they focus on a feature at a given depth (i.e. the centre book in Figure 1.2). This is effectively using a form of triangulation [4]. The second is by using the defocus of features on which our eyes are not focused, in order to estimate the depth, or distance, of those features (e.g. the two other books when focused on the centre book).

Hence, a relationship between defocus and depth is already established in the natural world. The task is to determine how successful we can be at using this effect to generate depth maps from 2-D images in the world of optics and image processing.

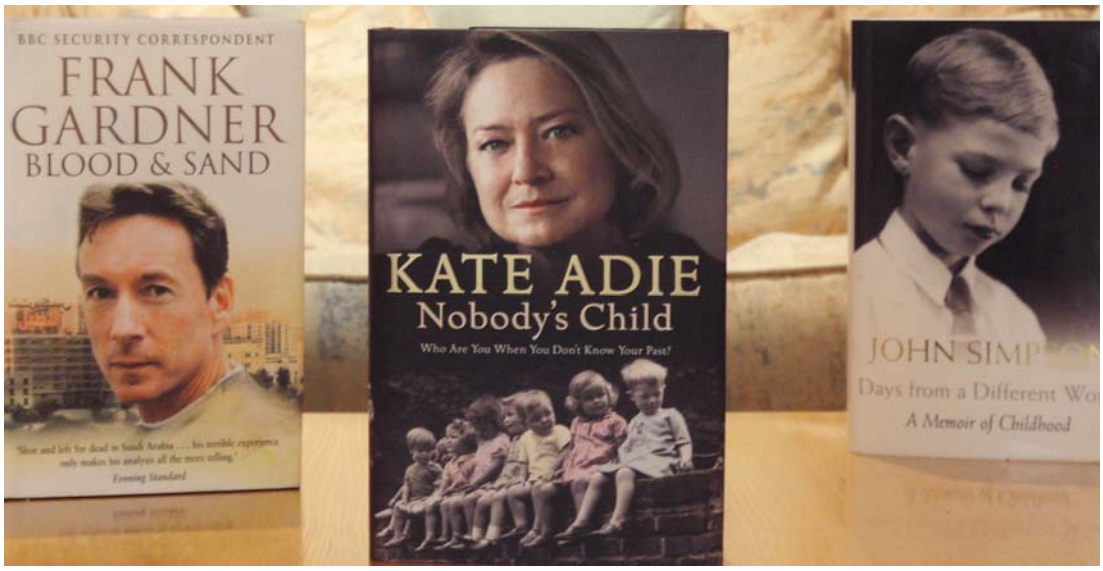


Figure 1.1 A digital image of three items at different depths.

Figure 1.1 shows three books on a table. The book in the centre is nearest to the camera lens, whilst the book on the left of the picture is approximately 30 cm further away from the lens, and the book on the right of the picture is approximately 60 cm further away from the lens. From the point of view of the image, all three books are reasonably in focus. The reason that some of the smaller writing on the book covers is difficult to read is mainly because of the resolution of the image.

However, from the human point of view, when looking at Figure 1.1 we do not actually absorb the entire view with our eyes in a fixed position, nor do we perceive the entire image to be in focus through a single fixed viewpoint. What actually occurs is that our eyes rapidly flit around the image, and only the feature that we are concentrating on at any given instant is actually in focus. However, our brain learns to 'stitch together' these multiple views, so that from our personal perspective we are simply looking at a total image that appears generally in focus.

The reader can test this theory by – for example – fixing their gaze at, say, the book in the centre of Figure 1.1. If the reader's eyes are fixed firmly on the book title, it

will be noticed that only the words of the title are clear and in focus, whilst everything else in the image is indistinct. It is under these conditions that our cognitive system will estimate the depth of the surrounding items from the amount of defocus they are subject to. It is difficult to exactly replicate this effect, but it appears similar to that shown in Figure 1.2.

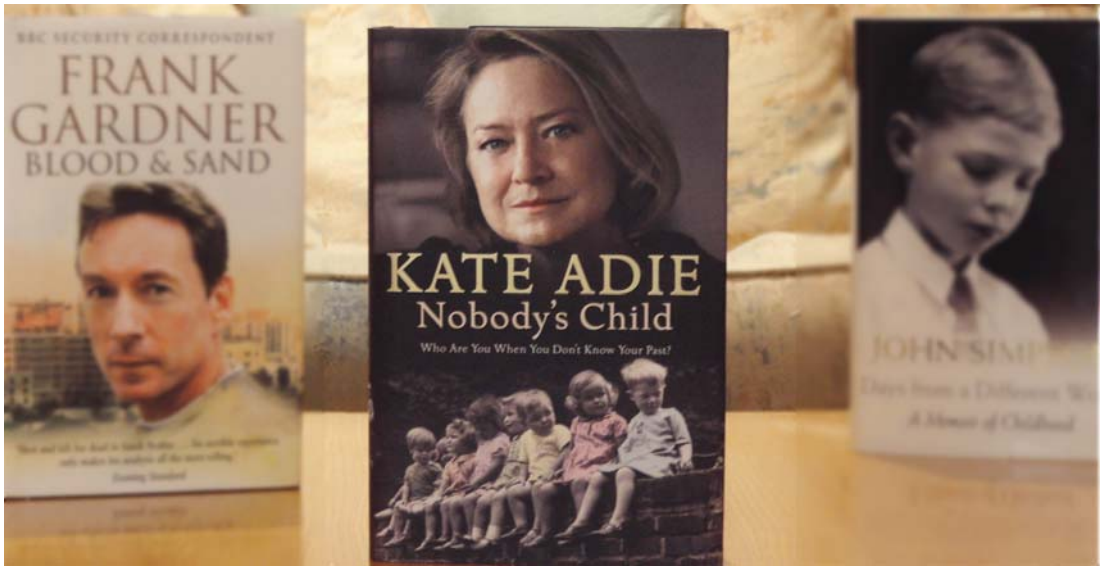


Figure 1.2 The effect of human vision with a fixed viewpoint

It can be seen from Figure 1.2 that everything else within the field of view, apart from the centre book, is quite badly defocused. Notice also that the book on the right of the image – being the furthest away from the lens – is defocused more than the book on the left. Hence our brain will automatically detect this and use it as a depth cue.

This clearly suggests that an ability to measure the degree of defocus related to an image feature would provide a measure of depth relative to that feature. This factor has been exploited by researchers with an interest in producing 3-D information from 2-D images. The basic theory that has enabled this effect to be exploited is discussed in Chapter 2 of this thesis (esp. section 2.1.1 Defocus/Depth Relationship), and the wide range of applications of this theory is reviewed in Chapter 3.

1.2 The structure of this thesis

The overall purpose of the thesis is to

- provide the reader with information on the fundamental concepts and theory of depth from defocus
- review and summarise the work of previous influential researchers in applying and developing this theory to formulate new DfD methods
- describe the work of this author in investigating and formulating a new technique for generating 3-D depth maps from a 2-D image.

A summary of the chapter contents appears in section 1.2.1 below.

1.2.1 Summary of chapter contents

Chapter 2 - Optics and DfD Methods

Provides the fundamental theory that relates lens systems and defocus/depth relationships. The necessity of modelling defocus blur – rather than assuming it is a simple radial spread of light intensity – is then discussed, and a typical defocus blur model is described. Finally, some common methods of using this theory to estimate depth from defocus are then described. (Note this latter section is – by definition – restricted to a few common methods, as chapter 3 reviews the greater range of methods adopted by other researchers).

Chapter 3 - A Summary of Influential Research in Depth-from-Defocus

Reviews a range of influential research into DfD methods that has occurred over the past twenty years. In order to provide better comparison, the chapter has been organised into groups of researchers who have adopted a similar type

of approach. These groups comprise *six* essential categories – those who have used either a single-image or a two-image approach – those who have used either passive illumination or active illumination – those who have analysed defocus effects in either the spatial domain or the frequency domain. Of course, there are varying combinations of these six categories used across the range of research reviewed, but the structure of this chapter helps to organise them into broadly similar types.

Chapter 4 - Investigating Depth Cues in a Projected Structured Light Pattern

Having chosen to investigate the use of a projected structured light pattern, this chapter describes the author's initial examination of such a pattern and the response of various pattern features to a change in depth. It also includes a description of the design of the initial programming algorithm required to examine such features in captured images. This program design forms the basis of the far more sophisticated programme that is eventually developed in the final stages of the project. Following analysis of the performance of a number of projected pattern features; the feature – or depth cue – that was considered to produce the best response[1.1] in relation to depth change is selected for further development.

Chapter 5 - The Development of an Active DfD System

This chapter describes how the selected depth cue is used in the development of a depth from defocus system. The development of the system is described in terms of both hardware configuration and software design. Novel features, such as the movement of the projected pattern during object analysis, and the

[1.1] The term 'best response' can be defined in this sense as being linear in relation to the change of depth, and being robust in terms of the ability of a programming routine to accurately resolve changes of the feature against depth change.

progressive development of 2-D, 3-D and finally 4-D look-up table algorithms in order to increase the number of depth estimates made across the area of the depth map are fully described. This latter development solves one of the remaining problems of depth from defocus techniques, i.e. that of spatial invariance. The accuracy of depth estimation, combined with spatial invariance, is detailed in the results presented in this chapter.

Chapter 6 - Applications of an Active DfD System

Describes the final modification made to the DfD system hardware arrangement in order to project the structured light pattern along the optical axis of the camera. This ensures that there are no areas of the camera's view of an object that are occluded from the pattern being projected. The chapter then proceeds to show the results of applying the DfD system to a number of objects that have increasing complexity in their 3-D shape. The purpose of this exercise is to test the system to its limits, and to illustrate the areas of improvement that a level of system design beyond these prototype stages would need to address in order to take the design towards a possible commercial system.

Chapter 7 – Conclusions and Further Work

Examines the results of the developmental work described in this thesis, and analyses how successful this work has been. Design features that are novel and innovative improvements over work completed by previous researchers are highlighted, and the contribution of these features to the improvement of DfD methods is discussed.

Recommendations are made for further developmental work that may overcome some of the limitations of the present system.

Appendix A – MATLAB Source Files

A selection of the main program and function files complete with comments.

Appendix B – Investigation Results Files

A collection of the results files from a number of experiments described in chapters 4 and 5. These include many images, etc., that indicate variations on which the results are based – but are too bulky to include in the chapter itself.

Appendix C – Equipment Specifications

Specification details for equipment used in the system prototype.

Chapter 1 – References

[1] Crane H; "A theoretical analysis of the visual accommodation system in humans"; *IEEE Trans. PAMI, v5, pp122-139*, 1993

[2] Born M, Wolf E; "Principles of Optics"; *London, Pergamon ; ISBN-13: 9780521642224;* 1966

[3] Ma Y, et al; "An Invitation to 3-D Vision – from images to geometric models"; *Springer-Verlag, New York, ISBN: 0-387-00893-4*; 2005

[4] Jain R, et al; "Machine Vision"; *McGraw-Hill; ISBN 0-07-0332018-7;* 1995;

Chapter 2

Optics and DfD Methods

The purpose of this chapter is to explain some of the fundamental theory that is used in Depth from Defocus (DfD) methods, and also that is specifically referred to in chapter 3 of this thesis where previous research in DfD is reviewed.

The range of methods (and related theory) applied to DfD applications by researchers to date has been enormous, as can be seen in chapter 3. Hence it is not possible to cover every eventuality in this chapter. For this reason the chapter has been designed to illustrate the main theoretical elements, and has been organised into four sections,

- Lens systems and defocus/depth relationships
- Modelling defocus blur
- Methods of using defocus to estimate depth
- A summary of other methods of measuring depth

2.1 Lens Systems and Defocus/Depth Relationships

It is common practice in DfD research to describe the principles of geometric optics in terms of thin lenses. The assumption is that, although the actual lens in use may be, for example, a proprietary compound lens, for the purpose of considering the effect of geometric optics as it applies to depth-from-defocus the model of a thin lens is adequate.

2.1.1 Thin lenses

A single lens that has only two refracting surfaces is known as a *simple lens*, and a lens that consists of a number of simple lenses is called a *compound lens* [1].

However, lenses can also be classified by whether they are *thin* or *thick* – i.e. whether or not the thickness of the lens is negligible in terms of its effect on the pathway of light waves passing through it. For the majority of discussion involving DfD optics it is appropriate to consider lenses to be *simple, thin* lenses in a *centred system* (i.e. all surfaces are rotationally symmetric about a common optical axis).

Centred spherical simple lenses can have various cross-sections dependent upon the desired refraction effect offered by the two surfaces. Figure 2.1 shows the main types that are available. The convention is to view these lens designs in terms of the light rays passing through the lens from left to right. Using this convention it can be seen that a *convex* or *positive* lens is thicker at the centre and tends to decrease the radius of the light wavefront to cause *convergence* of the light rays[2.1], whereas a *concave* or *negative* lens is thinner at the centre and tends to cause *divergence* of the light wavefront. This chapter will deal only with the biconvex type of spherical lens commonly used in DfD models.

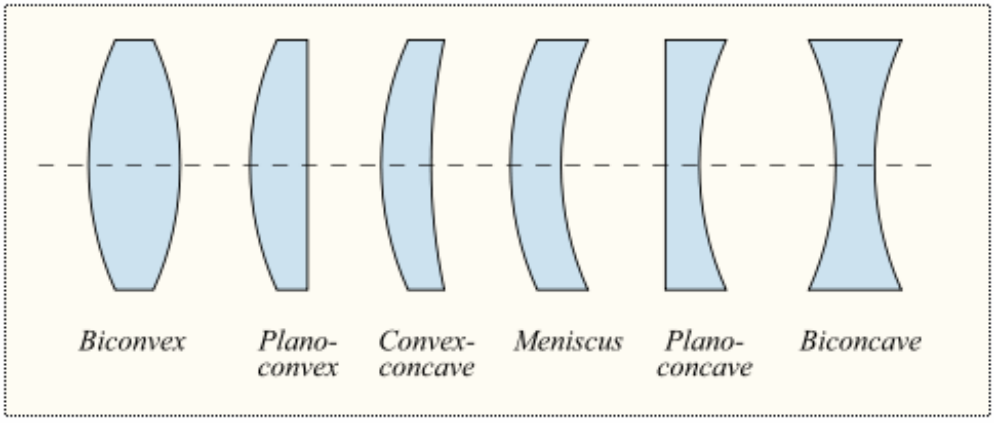


Figure 2.1 Cross sections of simple spherical lenses [2]

In order to understand thin-lens equations it is first necessary to have some knowledge of refraction at spherical surfaces. Figure 2.2 shows the refraction of a

[2.1] Assumes that the refractive index of the lens is greater than that of the media surrounding it.

wave from a point source *S* on a spherical interface of radius *R* centred at *C*. Point *V* is the *vertex* of the surface, and length $s_o = \overline{SV}$ is the *object distance*. At point *A* the ray $\overline{SA}$ will be refracted at the interface toward the optical axis (assuming refraction index $n_2 > n_1$), hence crossing the axis at *P* a distance from the vertex of $s_i = \overline{VP}$ (known as the *image distance*).

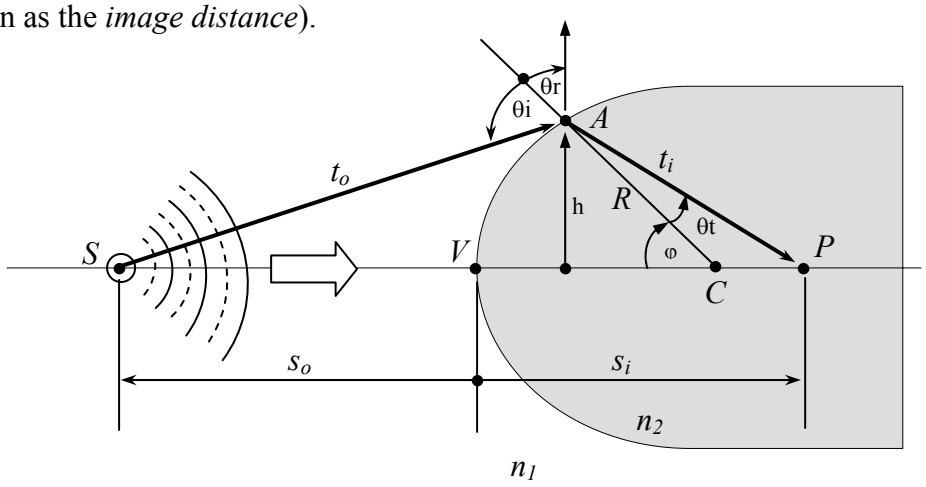


Figure 2.2 Refraction at a spherical interface[2.2]

Given that point *A* moves at the end of a fixed radius *R*, it can be shown (via Fermat's Principle [3]) that if *A* is moved to a different position on the surface by changing φ, then the new ray will not intercept the optical axis at *P*. To a first approximation (assuming relatively small values of φ, i.e. *A* is relatively close to *V*) this leads to the equation,

$$\frac{n_1}{s_o} + \frac{n_2}{s_i} = \frac{n_2 - n_1}{R} \qquad \{2.1\}$$

Where n_1 and n_2 are the refractive indices of the two mediums.

Points *S* and *P* in Figure 2.2 are known as the *conjugate points* and their location on the optical axis is determined by equation {2.1}. The *thin lens equations* can now be formed from equation {2.1}.

[2.2] Reconstructed from a diagram in [1]

When depth s_o is large for a fixed $(n_2 - n_1)/R$, s_i is relatively small. As s_o decreases, s_i moves further from the vertex, i.e. both θ_i and θ_t increase until eventually $s_o = f_o$ and $s_i = \infty$ (where f_o is the *front focal point* of the spherical interface). At this point equation {2.1} can be revised to,

$$\frac{n_1}{s_o} = \frac{n_2 - n_1}{R} \qquad \{2.2\}$$

so if depth s_o continues to decrease, s_i will have to become negative for equation {2.2} to still hold. This effect is illustrated in Figure 2.3.

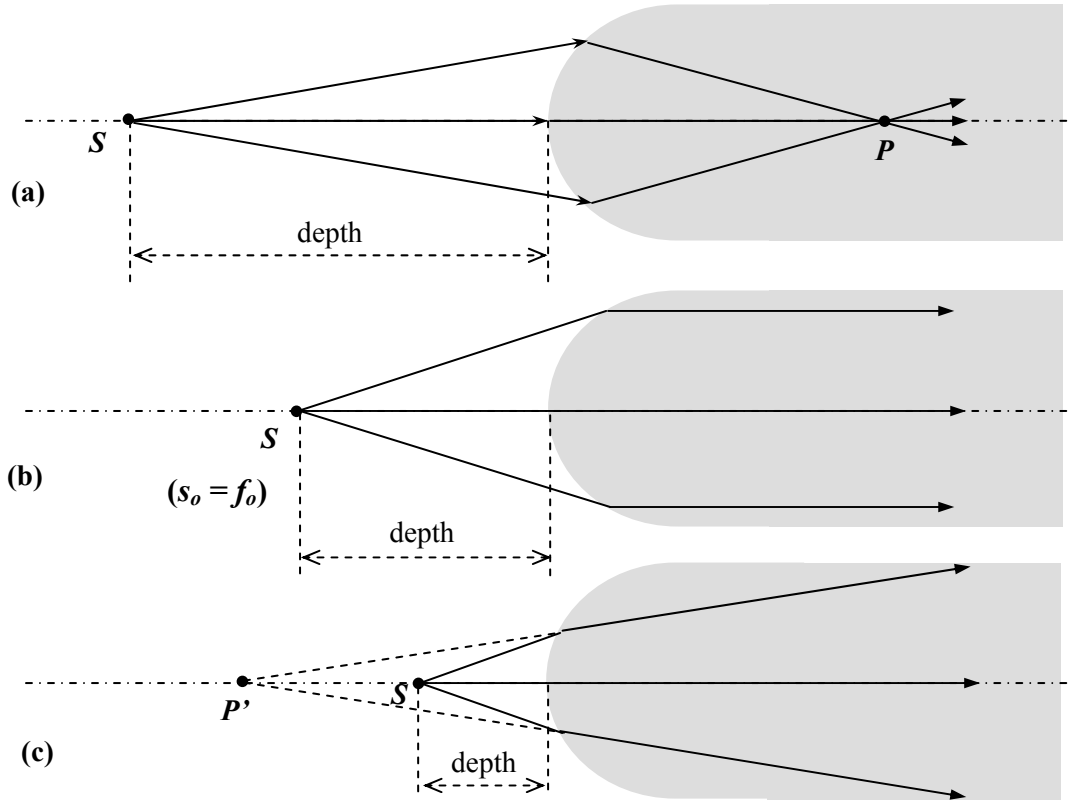


Figure 2.3 Refraction at a spherical interface for different source points *S*

Figure 2.3 (c) illustrates the so-called *paraxial rays* that meet on the optical axis at point *P'*. It can be seen in Figure 2.4 that when the single refractive surface of Figure 2.3 is converted to a double refractive surface – in the form of a *biconvex* lens

– then the paraxial rays are at the normal to the second surface. In effect, they represent the angle from which the second surface 'sees' the approach of rays from *S*. It can also be seen in Figure 2.4 that the paraxial rays issuing from *S* at distance s_{o1} meet at *P'* at a distance s_{i1} from V_1.

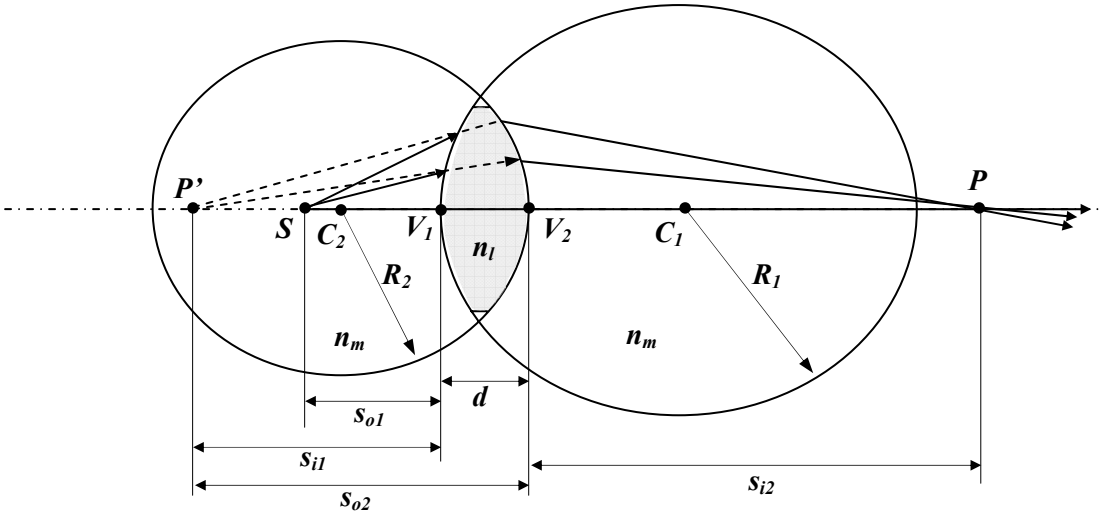


Figure 2.4 Spherical Lens Geometry

Equation {2.1} can now be restated in terms of the parameters in Figure 2.4, hence

$$\frac{n_m}{s_{o1}} + \frac{n_1}{s_{i1}} = \frac{n_1 - n_m}{R_1} \qquad \{2.3\}$$

As far as the second refractive surface in Figure 2.4 is concerned, point *P'* is effectively its object point at a distance s_{o2} away. As the second surface is in the medium of index n_1, it is taken that the object space for the second surface (including *P'*) has an index n_1.

Considering that $|s_{o2}| = |s_{i1}| + d$, then since s_{o2} is on the left (of surface 2) it is positive, so $s_{o2} = |s_{02}|$. Distance s_{i1} is on the left (of surface 1), and is therefore negative, $-s_{i1} = |s_{i1}|$. From this we have,

$$s_{o2} = -s_{i1} + d \qquad \{2.4\}$$

Hence, for the second surface we can restate equation {2.1} as

$$\frac{n_l}{(-s_{i1} + d)} + \frac{n_m}{s_{i2}} = \frac{n_m - n_l}{R_2} \qquad \{2.5\}$$

For equation {2.5}, $n_l > n_m$ and $R_2 < 0$ (with respect to surface 1), hence the right-hand side is positive. The addition of equations {2.3} and {2.5} gives

$$\frac{n_m}{s_{01}} + \frac{n_m}{s_{i2}} = (n_i - n_m)\left(\frac{1}{R_1} - \frac{1}{R_2}\right) + \frac{n_l d}{(s_{i1} - d)s_{i1}} \qquad \{2.6\}$$

However if the lens is assumed to be very thin ($d \to 0$) then the right-most term of {2.6} is effectively zero. A further simplification can be made by assuming the surrounding medium to be air ($n_m = 1$). This leads to the *thin-lens equation,*

$$\frac{1}{s_o} + \frac{1}{s_i} = (n_i - 1)\left(\frac{1}{R_1} - \frac{1}{R_2}\right) \qquad \{2.7\}$$

where $s_{o1} = s_o$ and $s_{i2} = s_i$. As $d \to 0$ the points V_1 and V_2 tend to merge, so s_o and s_i can be measured from the centre of the lens. In addition – as was the case with the single refractive surface of Figure 2.3 – if s_o is at infinity the image distance becomes the focal length f_i. Hence, it can be stated,

$$\lim_{s_o \to \infty} s_i = f_i$$
$$\lim_{s_i \to \infty} s_o = f_o$$

where f_i is the focal point on the image side of the lens, and f_o is the focal point on the object side of the lens (sometimes called the 'front' focal point).

For a thin lens (from equation {2.7}) it is evident that $f_i = f_o$, so the subscripts can be disregarded leaving,

$$\frac{1}{f} = (n_1 - 1)\left(\frac{1}{R_1} - \frac{1}{R_2}\right) \qquad \{2.8\}$$

and by relating equations {2.7} and {2.8} the final equation is the well known ***Gaussian Lens Formula*** that is commonly used in DfD research publications.

$$\frac{1}{s_o} + \frac{1}{s_i} = \frac{1}{f} \qquad \{2.9\}$$

2.1.2 Defocus/Depth Relationship

Development of the thin-lens equation {2.9} into an expression that relates defocus-to-object depth in an imaging system is well defined in [4], [5], and [6]. Figure 2.5 illustrates the geometry of this relationship.

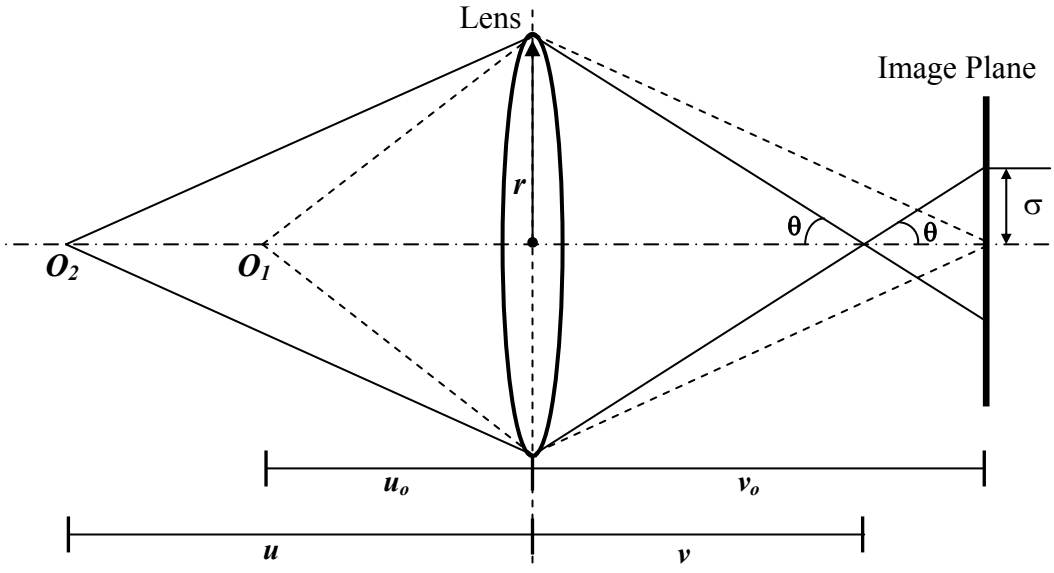


Figure 2.5 Imaging Geometry related to Defocus Blur

Figure 2.5 shows the light rays from an object point O_1 passing through a spherical thin lens of radius r at a distance u_o on the object side of the lens. As the rays from

this point converge exactly on the image plane at distance v_o from the lens, then object point O_1 is in focus. Object point O_2 at a distance u from the lens is not in focus with respect to the image plane as its rays converge at a point that is a distance v from the lens on the image side. It can be seen that the light rays from object point O_2 have diverged beyond the point of focus so that when they intercept the image plane the light energy is dispersed to form a defocused blur[2.3] of radius σ.

Given that the focal length of the lens is F, the parameters given in Figure 2.5 can be used in the thin lens equation {2.9} to form the relationship relative to object point O_2.

$$\frac{1}{u}+\frac{1}{v}=\frac{1}{F} \qquad \{2.10\}$$

This can be transposed to an expression that would provide the desired object distance u, hence

$$u=\frac{Fv}{v-F} \qquad \{2.11\}$$

As F is constant for a particular lens, and v can be fixed such that $v = v_o$, then it can be stated that a locus of points at distance $u = u_o$ will be in focus, i.e.

$$u_o=\frac{Fv_o}{v_o-F} \qquad \{2.12\}$$

For an object point where $u > u_o$ (e.g. point O_2 in Figure 2.5) and a blur circle is formed on the image plane, it can be stated that

$$\tan\theta=\frac{r}{v}=\frac{\sigma}{v_O-v} \qquad \{2.13\}$$

Renaming u as depth D, and combining equations {2.13} and {2.11}gives,

$$D=\frac{Frv_O}{rv_O-F(r+\sigma)} \qquad \{2.14\}$$

[2.3] The concept of defocus blur will be more fully explored in section 2.2 of this chapter.

Equation {2.14} can be simplified by using $f = F/Dia$, where f is the f-number of the lens[2.4]. Hence,

$$D = \frac{Fv_o}{v_o - F - 2\sigma f} \qquad \{2.15\}$$

Equation {2.15} can be used to estimate depth if the following factors are known,

- The focal length (F) of the lens
- The lens-to-image plane distance (v_o)
- The effective lens diameter (Dia)[2.4] to enable calculation of f.
- The blur circle radius (σ)

However, a limitation of equation {2.15} is the assumption that the blurring effect of the lens can be described by a simple blur circle of dispersed light. Section 2.2 of this chapter will discuss the implications of defocus blur in greater detail.

2.2 Modelling Defocus Blur

A blur circle is better described by the *point spread function* as defined below in section 2.2.1. However, a number of authors of DfD research papers have often used the notion of *line spread function* and *edge spread function* as a more practical means of estimating depth from defocus. These methods are defined in [1][4][7][11].

2.2.1 Point Spread Function

The point spread function is mainly the result of diffraction effects as the light passes through the lens. It can be thought of as the image brightness distribution produced by a point light source when the light flux (A_h) incident on the camera lens from the point light source is one unit. Hence,

$$A_h = 1 \qquad \{2.16\}$$

[2.4] With a proprietary lens, Dia is the effective diameter i.e. of the iris aperture that sets the f-number.

If incident light energy on the camera system from e.g. point O_1 (Figure 2.5) is b units, then the focused image can be described by $b\delta\left(x-\overline{x_h}, y-\overline{y_h}\right)$ where δ is the *Dirac delta function*[2.5][12]. This expression assumes that the Cartesian coordinate system on the image plane has its origin at $\left(\overline{x_h}, \overline{y_h}\right)$, such that

$$\overline{x_h} = 0 \text{ , and} \qquad \{2.17\}$$

$$\overline{y_h} = 0 \qquad \{2.18\}$$

Hence, $h(x,y)$ is effectively the response of the lens system to the input signal $\delta(x,y)$. On the assumption that a typical camera system has a circular aperture, the blurred image of the point light source is circular in shape, hence the term *blur circle*. For this case we can define the following parameters,

- r is the radius of the blur circle,
- R is the radius of the lens aperture,
- s is the distance between lens and image detector,
- q is a scaling factor defined by $q = r / R$
- f is the focal length of the lens

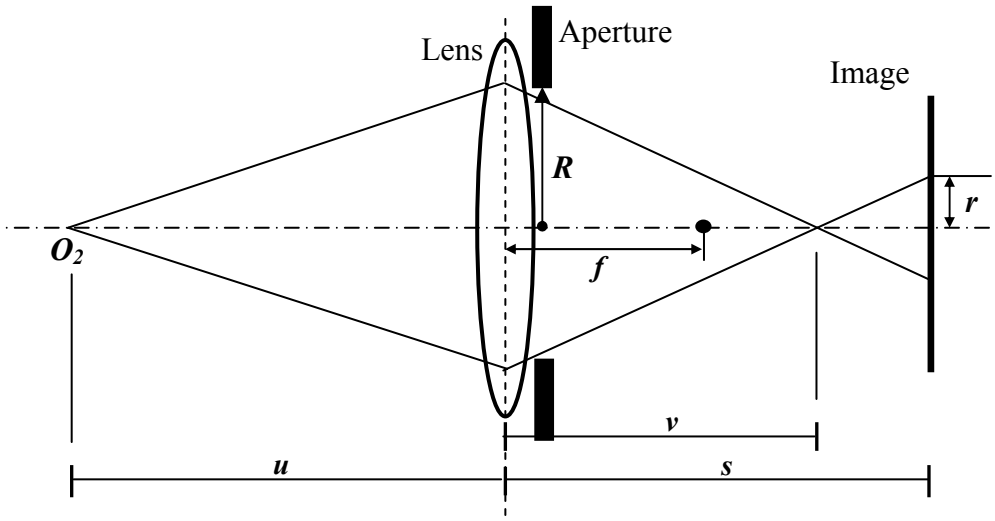


Figure 2.6 Camera Parameters for Defocus/Depth Relationship

[2.5] The Dirac delta function, often referred to as the impulse function, can informally be thought of as a function $\delta(x)$ where the area under the pulse (i.e. light flux energy) equals one.

From Figure 2.6 – using similar triangles – we have

$$\frac{r}{R} = \frac{s-v}{v} = s\left[\frac{1}{v} - \frac{1}{s}\right] \qquad \{2.18\}$$

Substituting for $\mathbf{1}/v$ by reference to the Gaussian Lens formula {2.9} gives

$$q = s\left[\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right] \qquad \{2.19\}$$

Hence,

$$r = Rq = Rs\left[\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right] \qquad \{2.20\}$$

It should be noted that q (and hence r) could be either positive or negative depending whether the image plane is *behind* or *in front* of the focused image of O_2. The sign of r cannot be determined from a single image, but this effectively gives ambiguity in determining the object distance. A method of avoiding this ambiguity is to set the distance between the image plane and the lens equal to the focal length (i.e. $s = f$). This causes q to always be negative, and enables a unique solution for the distance to be obtained.

Based solely on geometric optics, light intensity distribution within the blur circle is approximately constant. This model is generally known as the 'pillbox function'. Using equation {2.16) gives

$$h_1(x,y) = \begin{cases} \dfrac{1}{\pi r^2} & \text{if } x^2 + y^2 \le r^2 \\ 0 & \text{otherwise} \end{cases} \qquad \{2.21\}$$

However, due to lens diffraction, etc., an alternative model is commonly used for intensity distribution based on a two-dimensional Gaussian

$$h_2(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}} \qquad \{2.22\}$$

where σ is the spread parameter such that

$$\sigma = kr \text{ for } k > 0 \qquad \{2.23\}$$

k is a constant of proportionality characteristic for a given lens. It is determined experimentally as part of a calibration procedure.

Once a point spread function $h(x, y)$ for the lens is known (or assumed), the image $g(x, y)$ observed on the image plane is the result of a convolution between it and the focused image $f(x, y)$, hence

$$g(x, y) = h(x, y) * f(x, y) \qquad \{2.24\}$$

where $*$ denotes the convolution operator.

The point spread functions h_1 {2.21} and h_2 {2.22} are only two such examples. In order to deal with other forms of point spread function, the spread parameter σ_h[2.6] is used to characterise them. It can be shown that the spread parameter σ_{h1} relating to function h_1 is $r/\sqrt{2}$. From equation {2.20}

$$\sigma_{h1} = m_1 u^{-1} + c_1 \qquad \{2.25\}$$

where

$$m_1 = -\frac{Rs}{\sqrt{2}} \text{ and } c_1 = \frac{Rs}{\sqrt{2}}\left[\frac{1}{f} - \frac{1}{s}\right] \qquad \{2.26\}$$

Hence, for given parameters s, f, and R the spread parameter σ_{h1} depends linearly on the inverse of the distance u^{-1}.

Similarly, it can be shown that the spread parameter σ_{h2} of function h_2 is $\sqrt{2}\sigma$.

From equations {2.20} and {2.23}

$$\sigma_{h2} = m_2 u^{-1} + c_2 \qquad \{2.27\}$$

where

$$m_2 = -\sqrt{2}\,kRs \quad \text{and} \quad c_2 = \sqrt{2}\,kRs\left[\frac{1}{f} - \frac{1}{s}\right] \qquad \{2.28\}$$

[2.6] σ_h is the standard deviation of the distribution of the function h.

Again, for parameters ***k, s, f, R*** the spread parameter σ_{h2} depends linearly on the inverse of distance u^{-1}.

2.2.2 Line Spread Function

This represents the brightness distribution produced on the image plane by a line light source. In the following explanation it is assumed that the focused image of the line source lies along the *y*-axis. Under this condition the line source can be represented by ***δ(x)***. An illustration of the concept of line spread function can be seen in Figure 2.7.

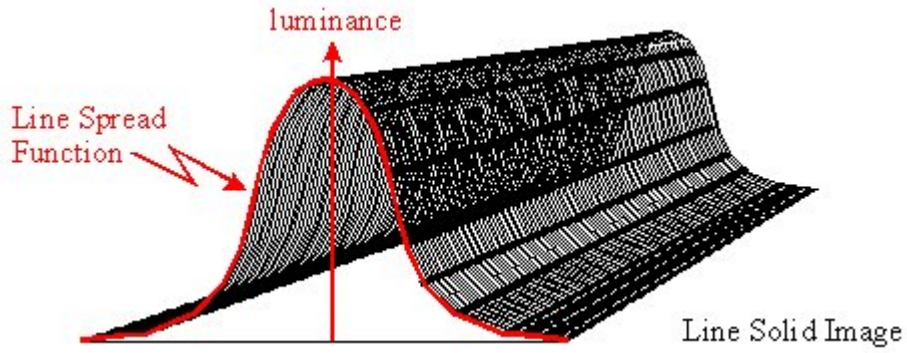


Figure 2.7 An Illustration of Line Spread Function [13]

For a circularly symmetric system blur, the line spread function ***l(x)*** (as shown in Figure 2.7) is the equation of the ***x***-section. Hence for all ***y***

$$l(x) = h(x, y) * \delta(x) \qquad \{2.29\}$$

It can be shown that the right-hand side of equation {2.29} can be expressed as a *line integral* along a line parallel to the y-axis. i.e.

$$l(x) = \int_{-\infty}^{\infty} h(x, y)dy \qquad \{2.30\}$$

The above two equations imply that ***l*** is a function of only ***x*** (i.e. is independent of ***y*** whatever the form of ***h(x,y)***). Following a similar process for that used in modelling

the point spread function in section 2.2.1, the parameters A_l, $\overline{x_l}$, and δ_l corresponding to $l(x)$ can be defined.

From equations {2.16} and {2.30},

$$A_l = 1 \qquad \{2.31\}$$

From equations {2.16}, {2.17}, {2.30} and {2.31},

$$\overline{x_l} = 0 \qquad \{2.32\}$$

The line spread function corresponding to the point spread function $h_1(x, y)$ from equation {2.21}, can now be defined using equation {2.30} as,

$$l_1(x) = \begin{cases} \dfrac{2}{\pi r^2}\sqrt{r^2 - x^2} & \text{if } |x| \le r \\ 0 & \text{otherwise} \end{cases} \qquad \{2.33\}$$

Relative to equations {2.22} and {2.23} the line spread function corresponding to h_2 can be determined to be

$$l_2(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\frac{x^2}{\sigma^2}} \qquad \{2.34\}$$

Note the requirement for only one variable x^2 in the power of e where $l_2(x)$ is the one-dimensional Gaussian line spread function.

2.2.3 Edge Spread Function

The edge spread function is closely related to the line spread function. However, instead of considering the image of an ideal line, the edge spread function of a system is the image of an ideal step function. This approach is more practical than the line spread function, because in most cases it is easier to manufacture a rectangular object which is placed in front of a background rather than an ideal line object. Also, in the case of a projected light patttern, it is difficult to produce a true

projected 'line' that appears infinitely thin on the object, whereas an 'edge' being a transition from one gray level to another can be produced with much more certainty. An illustration of edge spread function is shown in Figure 2.8

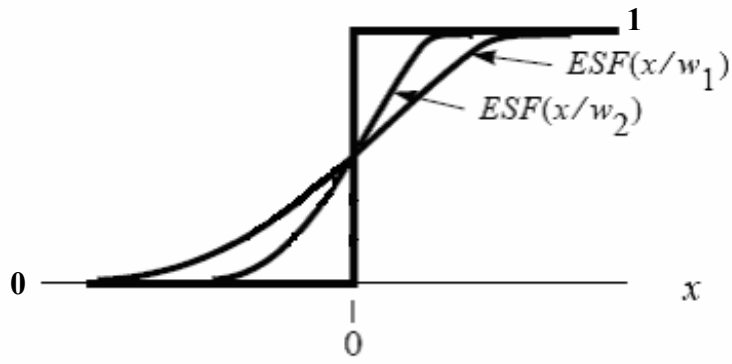


Figure 2.8 An edge after blurring with a generic PSF of two widths [14]

A unit step edge along the *y*-axis can be defined by the unit step function ***u(x)***. Hence, if ***e(x)*** denotes the edge spread function of an edge along the *y*-axis, then

$$\boldsymbol{e(x) = h(x, y) * u(x)} \qquad \{2.35\}$$

e(x) gives the image brightness distribution formed by an edge whose brightness is zero on the left of the *y*-axis and one (unit) to the right of the *y*-axis.

The edge spread function $\boldsymbol{e_1(x)}$ corresponding to $\boldsymbol{h_1(x, y)}$ is

$$\boldsymbol{e_1(x)} = \begin{cases} 1 & \text{if } \boldsymbol{x \geq r} \\ 0 & \text{if } \boldsymbol{x \leq -r} \\ \dfrac{1}{\pi r^2}\left[\pi r^2 - r^2 \operatorname{Cos}^{-1}\left(\dfrac{x}{r}\right) + x\sqrt{r^2 - x^2}\right] & \text{if } |x| < r \end{cases} \qquad \{2.36\}$$

Similarly, the edge spread function corresponding to $\boldsymbol{h_2(x, y)}$ from equation {2.22} is

$$\boldsymbol{e_2(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \int_{-\infty}^{x} e^{-\frac{t^2}{2\sigma^2}} dt} \qquad \{2.37\}$$

It can also be noted that the derivative of equation {2.37} is the point spread function. i.e. $\boldsymbol{\frac{de(x)}{dx} = psf}$

The defocus models described above – and many derivations of them – are commonly used in depth from defocus research methods for DfD systems using both passive light and active projected structured light patterns. The range of methods adopted varies widely, and most of the key researchers and their techniques are reviewed in chapter 3 of this thesis. However, section of 2.3 of this chapter will explain the theory of some important examples of depth from defocus methods.

2.3 Methods of using defocus to estimate depth

The author of this thesis has developed a depth from defocus method that essentially uses a *single image* to estimate scene depth (see chapters 4 and 5). However, it will be seen in chapter 3 of this thesis that the majority of previous work has used a *two-image* technique. The reason is that, in general, once an image of the object scene has been captured, measurement of the blur parameter σ in order to estimate depth via equation {2.27}, {2.34}, or {2.37} is problematic. This is because – with an image of an actual object – the data present in the image are the result of both the characteristics of scene features and those of the lens system.

One way of visualising this is to consider that an *out-of-focus sharp edge* on an object may look identical (in the image) to an *in-focus soft edge*[2.7]. This phenomenon would produce a similar estimate for the value of σ, and hence a similar estimate of depth for each feature – though they would actually be at different depths. A method of overcoming this problem is to only use object features such as *sharp edges* in order that the scene's contribution to the image is known (see [4]). However, this also leads to the major constraint that scene characteristics must be known prior to application of the method.

[2.7] The term 'soft edge' implies a rounded edge on an object in the scene.

The following sections 2.3.1, 2.3.2, and 2.3.3, describe methods of using two images in order to 'factor out' the contribution of the scene (as this contribution is the same in both images) and enable direct measurement of the degree of defocus.

2.3.1 Two-image method – using different size of aperture opening

Examples of this method are described in [4], [15], and [16]. A means of acquiring two images of the same scene that vary only in their depth of field (and, hence, focal gradient) is illustrated in Figure 2.9.

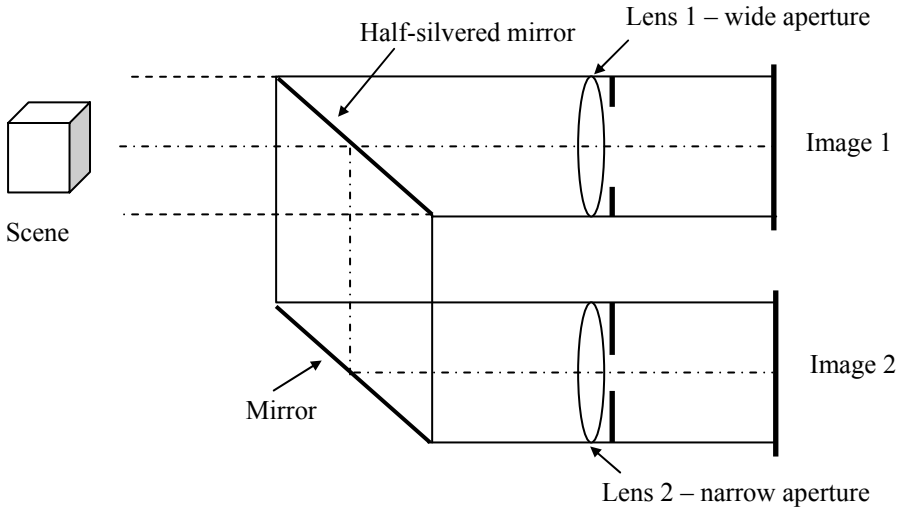


Figure 2.9 Arrangement to produce two identical images except for depth of field

It should be noted that the arrangement in Figure 2.9 may actually cause a variation in brightness of the two images, but the position of image features should be retained. Because the different aperture sizes cause different focal gradients, each image point will be focused differently in the two images. The distance between the viewer and the image point is essentially a function of the magnitude of the difference in focus.

The method used in [4] to obtain the degree of defocus at points in the images was to take a patch $f_1(r, \theta)$ centred at (x_o, y_o) within the first image $I_1(x, y)$

$$f_1(r,\theta) = I_1(x_o + r\cos\theta, y_o + r\sin\theta) \qquad \{2.38\}$$

and calculate its two-dimensional Fourier transform $\Im_1(t,\theta)$.

This is then repeated for patch $f_2(r, \theta)$ at the corresponding point in image two, giving $\Im_2(t,\theta)$. The assumption is made that the degree of blurring in the patch on each image is the result of a Gaussian point spread function $G(r, \sigma)$ convolved with the image patch $f_0(r,\theta)$ prior to blurring. The extracted image patches $f_1(r, \theta)$ and $f_2(r, \theta)$ will each be subject to a different spatial constant σ due to the difference in focal gradient caused by the different apertures.

Hence, the relationship of f_1 to f_2 (with spatial constants σ_1 and σ_2) is,

$$\frac{f_1(r,\theta)}{f_2(r,\theta)} = \frac{f_0(r,\theta) * G(r,\sigma_1)}{f_0(r,\theta) * G(r,\sigma_2)} \qquad \{2.39\}$$

It should be noted that equation {2.39} might have significant error when large amounts of defocus are present in the region of the image patch. The reason for this is that defocus blur from neighbouring regions may incur into the extracted patches f_1 and f_2 and corrupt the expected result of equation {2.39}.

Equation {2.39} can be used to derive a relationship between $\Im_1$ and $\Im_2$ (the Fourier transforms of the extracted image patches f_1 and f_2), and $\Im_0$ (the Fourier transform of the un-blurred image patch f_0). These transforms are derived in equation {2.40} below.

$$\begin{aligned} \Im_1(\lambda,\theta) &= \Im_0(\lambda,\theta) G\left(\lambda . \frac{1}{\sqrt{2\pi}\,\sigma_1}\right) \\ \Im_2(\lambda,\theta) &= \Im_0(\lambda,\theta) G\left(\lambda . \frac{1}{\sqrt{2\pi}\,\sigma_2}\right) \end{aligned} \qquad \{2.40\}$$

As only the amplitude of the transforms are of interest here equation {2.39} can now be simplified to,

$$\frac{\Im_1(\lambda)}{\Im_2(\lambda)} = \frac{\boldsymbol{G}\left(\lambda, 1/\sqrt{2\pi}\,\sigma_1\right)}{\boldsymbol{G}\left(\lambda, 1/\sqrt{2\pi}\,\sigma_2\right)} = \exp\left(\lambda^2 2\pi^2\left(\sigma_2^2 - \sigma_1^2\right)\right) \qquad \{2.41\}$$

where,

$$\Im(\lambda) = \int_{-\pi}^{\pi} \Im(\lambda, \theta)\, d\theta$$

Hence, σ_1 and σ_1 can be derived from $\Im_1$ and $\Im_2$ by taking the natural log of {2.41}

$$\lambda^2 2\pi^2\left(\sigma_2^2 - \sigma_1^2\right) = \ln \Im_1(\lambda) - \ln \Im_2(\lambda) \qquad \{2.42\}$$

In [4], equation {2.42} is formulated as a linear regression equation in λ^2, where

$$\boldsymbol{A} = 2\pi^2\left(\sigma_2^2 - \sigma_1^2\right)$$
$$\boldsymbol{B} = \ln \Im_1(\lambda) - \ln \Im_2(\lambda)$$

If one lens in the arrangement of Figure 2.9 is configured as a pinhole camera, and hence $\sigma_1 = 0$, then $\boldsymbol{A} = 2\pi^2\sigma_2^2$ and equation {2.15} – derived in section 2.1 – can be used to solve for depth, e.g.

$$\boldsymbol{D} = \frac{\boldsymbol{F}\boldsymbol{v}_0}{\boldsymbol{v}_0 - \boldsymbol{F} - 2\sigma_i \boldsymbol{f}_i} \qquad \{2.43\}$$

where $\boldsymbol{f}_i$ is the *f*-number of the lens system.

2.3.2 Two-image method – using two image detectors and telecentricity

This method uses a mechanism by which the required distance between the lens and the scene object is constant, but – on the image side – the distance between the lens and the image detector is varied along the optical axis. In some cases, the same effect is achieved by using two image detectors set at different distances from the lens,

using a beam-splitter in order to ensure that each image detector 'sees' the same image from the lens.

This technique is capable of producing two images of the same scene with different degrees of defocus as required, but it does introduce a problem not present in the 'variable aperture' method. The variation in image detector to lens distance will also produce a zoom effect for all image points that are not on the optical axis. This effect is illustrated in Figure 2.10. This means that – without correction – the two images are spatially variant, and hence unusable for application of the DfD methods described above. A method of preventing the zoom effect whilst retaining the variation in defocus is to include a ***telecentric*** lens arrangement. The effect of this is illustrated in Figure 2.11. An example of this method – using telecentricity and two image detectors at different distances – can be seen in [10] and is described below.

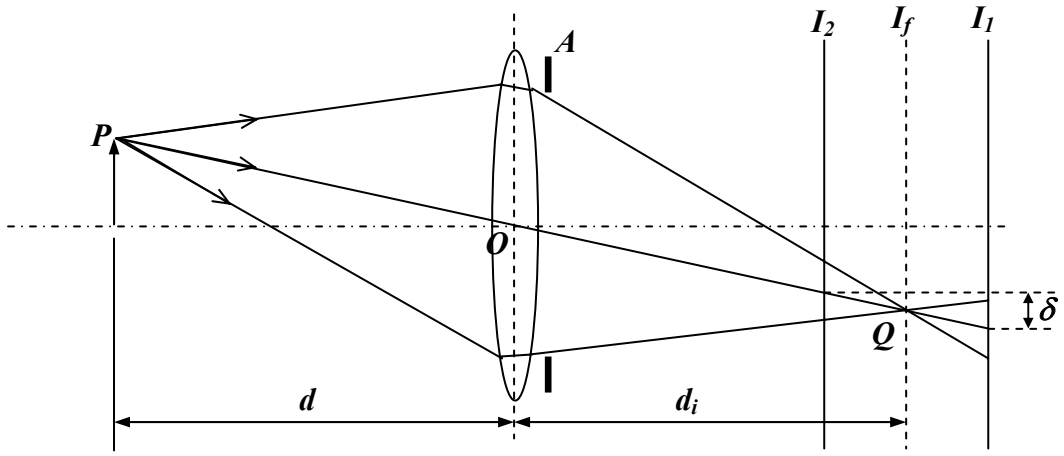


Figure 2.10 Spatial invariance with two image planes

Figure 2.10 shows the effect of two image detectors that are positioned at different distances from the lens. The principal ray path of a scene object point ***P*** passes through the centre of the lens at point ***O*** and – via a beam-splitter arrangement (not shown in Figure 2.10) – reaches each image detector. However, because the principal ray diverges from the optical axis as it leaves the lens, there is a natural

zoom effect between image planes I_2 and I_1 where all source object points will spread outwards radially. The variation in spatial position δ increases in proportion to the distance between I_2 and I_1.

The outer-most rays emanating from object point ***P*** are bounded by lens aperture ***A***. It can be seen that the 'spread' of these rays will produce a defocus blur on both image planes that are positioned either side of the focal point ***Q***. Before analysis to determine an estimate of depth ***d*** can begin, a method of correcting the spatial shift must be devised.

Converting a lens to a telecentric arrangement will retain the defocus features seen in Figure 2.10, but will eradicate the zoom effect. Effectively it means that both image detectors 'see' exactly the same scene through the lens, and only the focal gradient is different. Figure 2.11 illustrates how telecentricity is achieved.

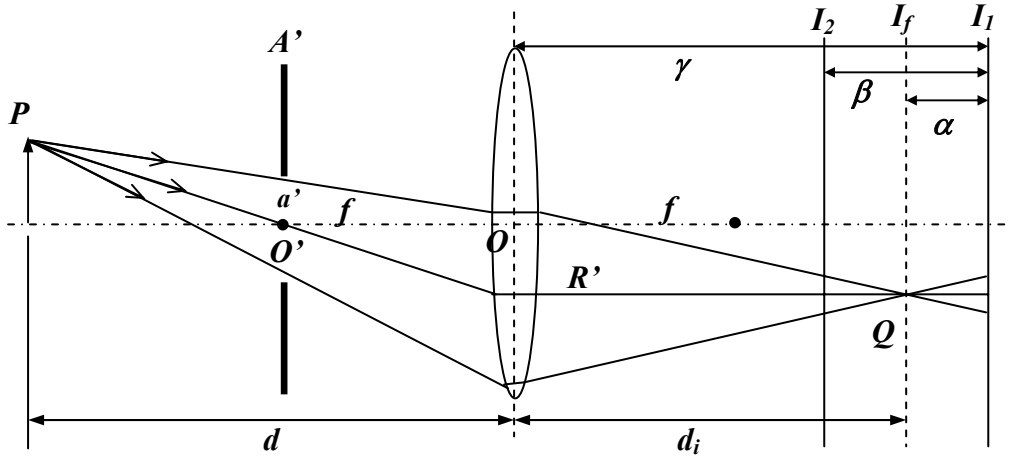


Figure 2.11 Constant magnification using an aperture at the lens front-focal plane

Figure 2.11 shows an aperture ***A'*** that is fitted externally to the lens arrangement and is positioned precisely on the front-focal plane of the lens. Geometrical analysis can show that the principal ray of light ***R'*** emitting from any scene point and passing through the centre ***O'*** of aperture ***A'*** will emerge parallel to the optical axis on the image side of the lens. This means that each scene point will retain exactly the same

position in the image planes of I_1 and I_2, but the relative blur effect between the two images is also still retained.

In order to use the two images from I_1 and I_2 to estimate depth, the authors (Nayar, Watanabe, Noguchi) [10] produced a form of Laplacian focus operator that was tuned to the optimised structured pattern projected onto the scene. This tuned focus operator was applied to both images in order to obtain focus measure images $g_1(x, y)$ and $g_2(x, y)$. Given that the structured pattern used in this case had a single dominant frequency ($1/t_x$, $1/t_y$), then

$$\frac{g_1(x,y)}{g_2(x,y)} = \frac{G_0\left[\frac{1}{t_x}, \frac{1}{t_y}; \alpha\right]}{G_0\left[\frac{1}{t_x}, \frac{1}{t_y}; \alpha - \beta\right]} \qquad \{2.44\}$$

Where G_0 is the focus measure of the discrete image expressed in Fourier domain.

It is shown in [10] that the only factor in G_0 affected by parameter α is the defocus function H. Hence, equation {2.44} can be re-expressed as

$$\frac{g_1(x,y)}{g_2(x,y)} = \frac{H\left[\frac{1}{t_x}, \frac{1}{t_y}; \alpha\right]}{H\left[\frac{1}{t_x}, \frac{1}{t_y}; \alpha - \beta\right]} \qquad \{2.45\}$$

As a measure of defocus, equation {2.45} is not bounded, which can be problematic from a computational point of view. This is remedied in [10] by using a normalisation, e.g.

$$q(x,y) = \frac{g_1(x,y) - g_2(x,y)}{g_1(x,y) + g_2(x,y)} = \frac{H\left[\frac{1}{t_x}, \frac{1}{t_y}; \alpha\right] - H\left[\frac{1}{t_x}, \frac{1}{t_y}; \alpha - \beta\right]}{H\left[\frac{1}{t_x}, \frac{1}{t_y}; \alpha\right] + H\left[\frac{1}{t_x}, \frac{1}{t_y}; \alpha - \beta\right]} \qquad \{2.46\}$$

Figure 2.12 shows that q is a monotonic function of α[2.8] and hence can be used directly to estimate depth by use of the lens law {2.9}.

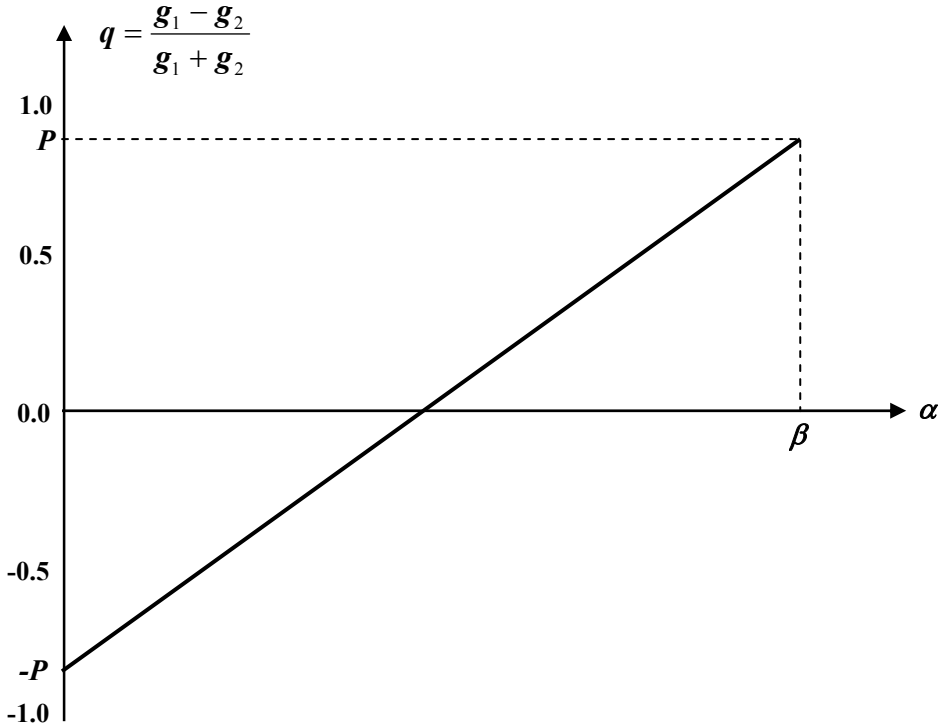


Figure 2.12 Relationship between focus measures g_1 and g_2 and defocus measure α

2.3.3 Two-image method – changing focal length

This method will not be described in extensive detail here as it is rarely used in practice. Although it represents the third possible alternative means of producing two images of the same scene with different focal gradients, it is difficult to implement as it introduces a zoom effect. However, unlike the two-image method where the distance between lens and image detector is varied, this zoom effect cannot be corrected by a telecentric arrangement, as this is itself dependent upon a fixed focal length of the lens. It also has no particular advantage over the previous two methods discussed. An example of this technique is given in [16] where the lens arrangement shown in Figure 2.13 is suggested.

[2.8] In practice this is not always true where the frequency of the texture pattern is high.

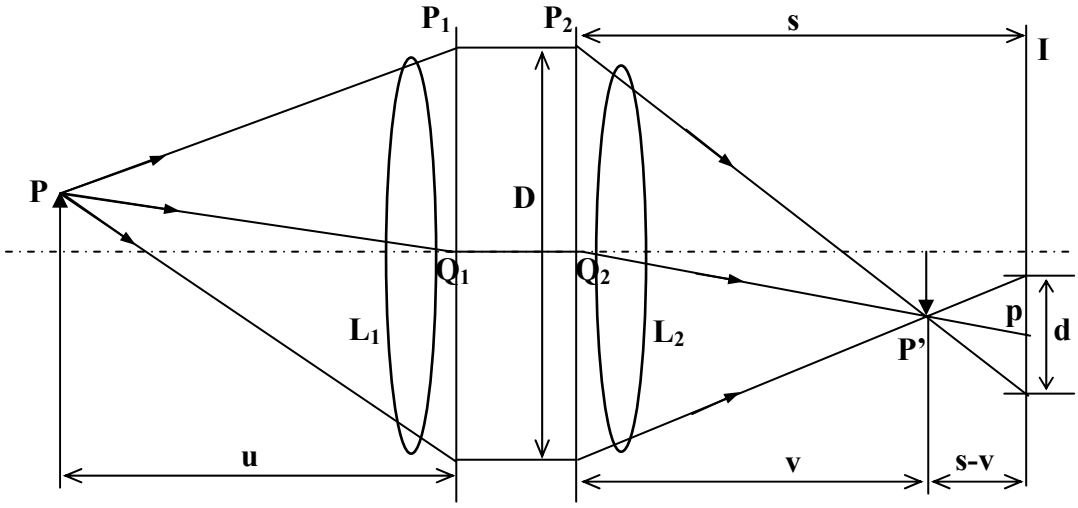


Figure 2.13 Geometry of two-lens system for variable focal length

For Figure 2.13, $\mathbf{P_1}$ and $\mathbf{P_2}$ are the first and second principal planes, **P** is the object point, **p** is the corresponding image point, **I** is the image detector plane, **s** & **D** are camera parameters, **u** is the object distance, **v** is the focused image distance, **d** is the blur circle diameter, and $\mathbf{Q_1}$, $\mathbf{Q_2}$ the first and second principal points.

The focal length of the lens can be changed by moving lens $\mathbf{L_2}$ relative to $\mathbf{L_1}$ and **I**. The front lens $\mathbf{L_1}$ remains fixed relative to the object point **P**. The purpose of this is to minimise the potential correspondence problem whereby a zoom effect would be introduced if $\mathbf{L_1}$ were to be moved.

The effective focal length of this arrangement is determined by equation {2.47}

$$\frac{1}{f} = \frac{1}{f_a} + \frac{1}{f_b} - \frac{l}{f_a f_b} \qquad \{2.47\}$$

where f_a and f_b are the focal lengths of the two lenses, and l is the distance between the two lenses. In order to use this method the positions of the principal planes and principal points must be determined via geometric optics.

2.4 Other methods of measuring depth

This section will summarise a number of other methods that may be used to calculate the distance of various points in the scene relative to the position of the camera lens.

2.4.1 Stereo imaging [17]

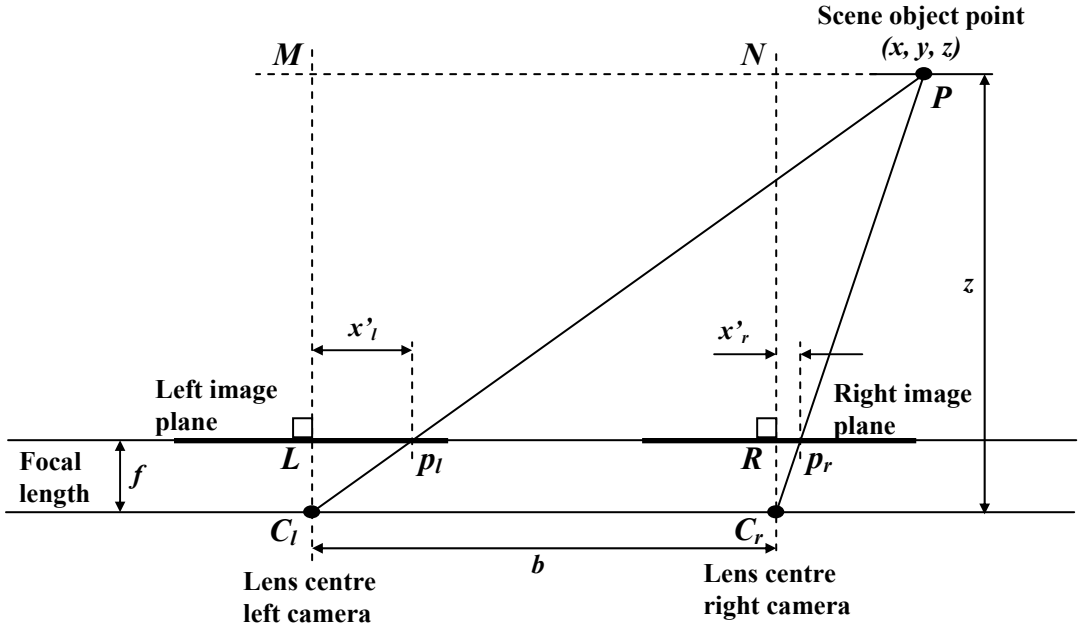


Figure 2.14 Binocular Stereo Geometry

Figure 2.14 illustrates the principle of stereo imaging using the model of binocular[2.9] stereo geometry. The scene point $\boldsymbol{P}$ is detected at points $\boldsymbol{p_l}$ and $\boldsymbol{p_r}$ in the left and right image planes, respectively. The required depth z can be calculated by the method of triangulation using known parameters from the geometry of the stereo arrangement.

Triangles $\boldsymbol{PMC_l}$ and $\boldsymbol{p_lLC_l}$ are similar, giving

$$\frac{x}{z} = \frac{x'_l}{f} \qquad \{2.48\}$$

Triangles $\boldsymbol{PNC_r}$ and $\boldsymbol{p_rRC_r}$ are also similar, giving

$$\frac{x-b}{z} = \frac{x'_r}{f} \qquad \{2.49\}$$

[2.9] 'Binocular' implies that the two image sensors are mounted on the same image plane, and that their optical axes are parallel. Other forms of optical stereo use systems where the optical axes of the two sensors converge at a distance from the image plane. The principle of depth estimation is similar.

Combining equations {2.48} and {2.49}, gives

$$z = \frac{bf}{\left(x'_l - x'_r\right)} \qquad \{2.50\}$$

Hence the depth at various scene points can be estimated by knowing the disparity (i.e. $x'_l - x'_r$) between corresponding image points. Stereo imaging has been widely used in many optical systems that estimate depth, and is capable of providing high levels of accuracy (typically <2% error). However, its main limitation can be seen from Figure 2.14 and equation {2.50}, where parameters f and b will be known *a priori*, but image points x'_l and x'_r must be established, and this requires that they are recognised in each image as a pair of conjugate points representing the same scene point. This is often referred to as the 'correspondence problem', and is one of the main limitations that can be avoided by methods such as depth from defocus (as reported in this thesis).

2.4.2 Photogrammetry

Photogrammetry is related to stereo imaging in the sense that it uses similar triangulation techniques based on the geometric properties of the imaging devices. However, it differs from general stereo imaging in that – in order to overcome the 'correspondence problem' – it specifically uses known targets that can readily be identified in the scene. When photogrammetry is used for e.g. topographic land mapping, measurement of architectural features, etc., common points can be identified in the scene. For applications such as the measurement of manufactured artefacts, it is common practice to fix target devices onto the artefact being measured (see Figure 2.15). The software used to analyse the images can be designed to identify the targets in each image, calculate the disparity between the image points, and hence calculate the scene coordinate points of the target.

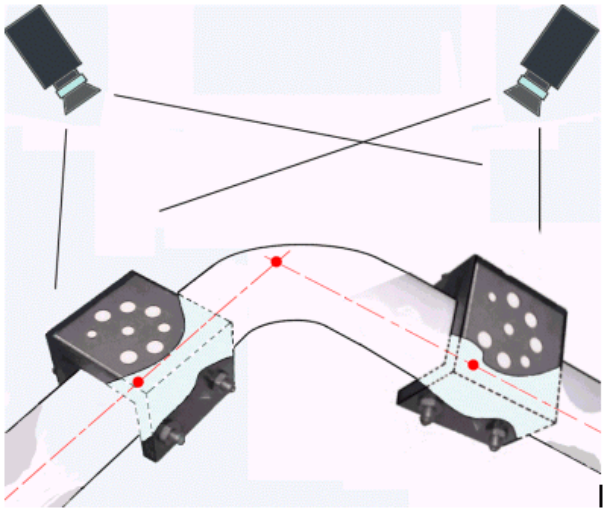


Figure 2.15 Coded photogrammetry targets fitted to a manufactured pipe

Photogrammetry can use multiple camera positions in order to increase the accuracy of its scene coordinate estimations. It is particularly suited to the measurement of large manufactured objects, e.g. aircraft wings.

2.4.3 Light stripe projection

A projector (typically a laser light projected through a cylindrical lens) scans a light stripe across an object, and a camera positioned at an angle to the projector images the light stripe as the shape of the object deforms it. The system is calibrated such that software analysing the image can estimate the depth of the object feature from the amount of stripe displacement (see Figure 2.16).

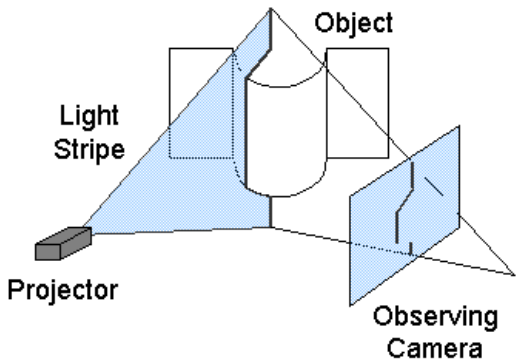


Figure 2.16 The Principle of Light-Stripe Projection [18]

2.4.4 Laser tracker

Laser tracker devices are also suitable for measuring large-scale objects. They are based on the combination of two techniques

1. A laser interferometer – capable of measuring relative distance
2. An optical encoder – capable of measuring the azimuth and elevation of a beam-steering mirror.

The principle of the laser interferometer is illustrated in Figure 2.17.

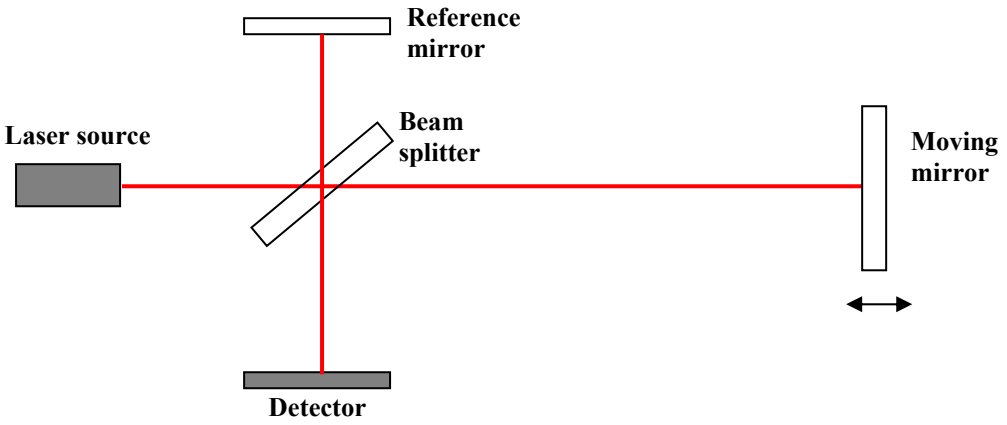


Figure 2.17 Laser Interferometer Arrangement [19]

The coherent laser light is split into two beams, one forming a reference, whilst the other is reflected back from a mirror or similar reflective device that is positioned some distance away. The reflected beam merges with the reference beam producing interference. The number of interference fringes will increase as the mirror is positioned further away. These can be detected and counted to provide a linear measurement of distance (or depth).

To enable the system to 'track' movement of the target other than in a straight line, a beam-steering mirror directs the laser beam to a retro-reflective target (Figure 2.18). If the beam is aimed directly at the centre of the retro-reflective target it will reflect the beam back along the same path. However, as the target changes its lateral position, the beam will be reflected with a parallel offset. A two-dimensional sensor

can measure this offset and cause the laser tracker to adjust the beam-steering mirror until the beam is once again aligned with the centre of the target. Laser trackers are used in a variety of measurement and alignment applications.

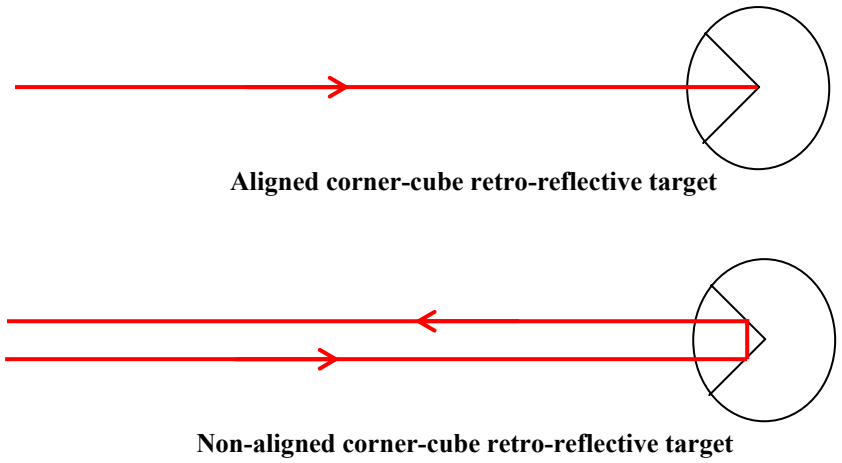


Figure 2.18 Laser-tracker Retro-Reflective Tracking Mechanism [19]

2.4.5 Laser time-of-flight

Laser time-of-flight sensors can measure large depths typically ranging from a few meters to several kilometres. For the shorter distances in this range they can often measure distance to a single selected point without using a reflective target, but for distances in the kilometre range a reflective target would normally be used. Generally, a laser light beam (modulated to a wavelength of 20 m to 30 m) is transmitted in a short, timed pulse. By comparing the phase shift between the transmitted and reflected wave the time-of-flight can be determined, and if the speed of light is known the distance can be estimated. Normal practice is to time a number of pulses and determine the average. Hence, a trade-off between accuracy and speed of measurement occurs. These instruments can be very compact, and are often used in conjunction with a sighting device to target the beam onto the selected object point.

2.4.6 Ultrasound sensors

Ultrasound sensors work on a similar principle to time-of-flight lasers except they emit a short burst (or 'ping') of sound at ultrasonic frequency. This reflects off the object surface and the sensor's receiver uses the known speed of sound to calculate the time taken for the return journey, and hence estimates the distance. An advantage of ultrasound sensors is that they can be used to detect the distance to almost any type of surface (including glass or liquid). A disadvantage is that the speed of sound is variable, and is particularly dependent upon the air temperature. A method of measuring the air temperature and compensating the depth estimate for this variable is normally included.

2.4.7 Coordinate Measuring Machines (CMM)

These devices use a very sensitive electro-mechanical touch probe (Figure 2.19a).

a. **b.** **c.**

Figure 2.19 Coordinate Measuring Machine Devices and Applications [20]

The probe is mounted on a multi-axis machine that allows the probe stylus to be manoeuvred into a range of positions. This allows sensitive contact to be made at numerous points on the surface of a manufactured artefact of relatively complex 3-D shape. Dedicated software, programmed to suit the application being measured, will receive the 3-D coordinates of each contact point made by the probe and produce a 3-

dimensional map of the object. CMM machines can vary widely in their design, from small portable arms capable of measurements within a small volumetric range (e.g. small manipulated tubes – Figure 2.19c), to large bridge-type machines that may have a volumetric range of several cubic meters (i.e. that could measure a vehicle – Figure 2.19b). Disadvantages of CMM designs includes

- The necessity to physically probe the device – can be problematic if the device has a soft surface or is made of a flexible material
- Error in the machine axis sensors that causes incorrect coordinate readings to be recorded at the contact points – this often has to be corrected by regular re-calibration and, e.g. the updating of error-mapping software.
- With regard to the analytical software, the CMM must be set up to suit the specific application being analysed.

An advantage of the CMM for many users is that it is relatively robust in an industrial environment, and it can cope with variations in size and complexity of shape more readily than many optical systems (unless they are very sophisticated).

2.5 Summary of Chapter 2

This chapter has explained how light rays from a point source of light travel through a refractive surface. This was then developed to show how spherical lenses are used in basic imaging systems. In particular, the thin lens equations were devised, and this led to the Gaussian Lens Formula {2.9}. The Gaussian Lens Formula forms the basis of imaging geometry, and this was explained in relation to the creation of blur effects on the imaging receptor.

Methods of modelling image blur were discussed in section 2.2, where the concepts of *point spread function*, *line spread function*, and *edge spread function* were explained.

In section 2.3, a number of methods for estimating depth from defocus that related imaging geometry to defocus models were described. The examples used in section 2.3 were only a small sample of the enormous range of DfD techniques used by researchers to date, but their reason for inclusion in section 2.3 was simply to provide explanation of the relationship between imaging geometry, defocus blur models, and depth from defocus techniques. A review of all areas of key research in depth from defocus will be given in chapter 3.

Finally, in section 2.4, a summary of other methods of measuring depth was given. This included optical methods such as stereo imaging, photogrammetry, and laser-based methods. It also dealt briefly with non-optical methods such as ultrasound sensors and coordinate measuring machines. A number of advantages and disadvantages were given for each of these methods.

Chapter 2 – References

[1] Hecht E; "Optics – 4th ed"; *Addison Wesley Longman Inc; ISBN 0-8053-8566-5;* 2002.

[2] http://en.wikipedia.org/wiki/Lens_(optics)#Types_of_simple_lenses(30-08-07)

[3] http://www.physics.nmt.edu/~raymond/classes/ph13xbook/node36.html (30-08-07)

[4] Pentland A P, "A New Sense for Depth of Field", *IEEE Trans. Pattern Analysis and Machine Intelligence*, *v9, No4, pp.523-531*, July 1987.

[5] Ens J and Lawrence P, "A Matrix-Based Method for Determining Depth-from-Defocus", *IEEE PAMI, pp600-606*, 1991

[6] Ens J and Lawrence P, "An Investigation of Methods for Determining Depth from Defocus", *IEEE Transactions on Pattern Analysis and Machine Intelligence; v15, No2, pp97-108*, 1993

[7] Surya G, Subbarao M, "Depth from Defocus and Rapid Autofocusing: A Practical Approach", *IEEE Computer Vision and Pattern Recognition*, *v13. No3, pp61-67***,** 1993

[8] Pentland A P, Scherock S, Darrell T, Girod B, "Simple range cameras based on focal error", *J.Opt.Soc.Am A; v11 No11; pp.2925-2934*, 1994

[9] Xu S, et al, "Range measurement from defocus gradient", *Machine Vision and Applications; v8; pp179-186*, 1995

[10] Nayar SK, Watanabe M, Noguchi M, "Real-time Focus Range Sensor", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *v18, No12, pp1186-1197*, 1996

[11] Petrou M, Bosdogianni P, "Image Processing", *John Wiley and Sons Ltd, ISBN 0-471-99883-4,* 1999

[12] http://mathworld.wolfram.com/DeltaFunction.html (03-09-07)

[13] http://www.yorku.ca/eye/spread.htm (04-09-07)

[14] http://coen.boisestate.edu/EBarneySmith/Papers/icdar_2001.pdf (04-09-07)

[15] Pentland A, Darrell T, Turk M, Huang W; "A Simple Real-Time Range Camera"; *CH2752-4/89/0000/0256$01.00 IEEE; pp256-261*; 1989

[16] Subbarao M; "Parallel Depth Recovery by changing Camera Parameters"; *Proc.Int.Conf.on Computer Vision; Tarpon Springs FL, IEEE; pp.149-155*; 1988

[17] Jain R, et al; "Machine Vision"; New York : *McGraw-Hill, ISBN 9780071134071*; 1995

[18] http://www.ifp.uni-stuttgart.de/forschung/optical/dense_surf/dense_surf.htm (Sept 2007)

[19] http://www.optical-metrology-centre.com/Downloads/Tech_Briefs/TechBrief_LaserTracker.pdf (Sept 2007)

[20] http://www.itpgroup.co.uk/products/new_cmms/new_cmms.html (Sept 2007)

Chapter 3

A Summary of Influential Research in Depth-from-Defocus

Any researcher carrying out a detailed investigation of published work on the subject of depth-from-defocus (DfD), would almost certainly come to the conclusion at a very early stage that Alex P Pentland's paper [1] entitled 'A New Sense for Depth of Field' is widely accepted as the seminal paper representing the first serious work in this topic. This chapter will summarise and briefly evaluate the work of Pentland and other key researchers who have collectively pursued an extremely wide range of techniques in their attempts to extract depth information from defocus effects in 2-dimensional images.

As the variation in methods of extracting 3-D information from image defocus has been so diverse over the past 20 years, this chapter has been organised to present researchers in broadly similar groups. These will be defined by whether the research approach has used,

- Single-image or double-image
- Passive or active illumination
- Analysis of the depth information in the spatial or frequency domain

It will be seen from tables (below) summarising the key outcomes of this work, that the majority of useful results to date have been achieved via multiple-image techniques. Whereas the work in later chapters of this thesis presents robust results derived from a technique that requires only a single-image.

3.1 Single-image with Passive Illumination Methods

Previous Work Using SINGLE IMAGE with PASSIVE Illumination Methods						
Author Reference	**Method**	**Spatial or Frequency Analysis**	**Assumed Defocus Operator**	**Spatial Resolution**	**Depth Range**	**Error**
Pentland [1] 1987	Uses sharp edges in image – hence rate of intensity change due mainly to system psf	Spatial *variant*	Gaussian	Feature Dependant	---	---
Cardillo J, et al [5] 1990	Sharp edge (similar to [1]) – transforming camera coordinates to world coordinates	Spatial *variant*	Gaussian	Feature Dependant	750mm	1.9%

Table 3.1

Having realised that a correlation can be found between the degree of blur in an image feature and its distance from the lens, Pentland [1] separated characteristics of the scene (i.e. soft edges generating a natural intensity gradient) from those of the lens system itself (the point spread function (psf) of the lens system), by analysing the effect on sharp edges in the image. He used a convolution of image data *I(x, y)* representing the sharp edge with that of a Gaussian psf [represented by $G(r, \sigma)$, where $\boldsymbol{r}$ is the blur radius, and $\boldsymbol{\sigma}$ is a measure of the 'spread' caused by the psf], and then defined the value ***C(x, y)*** as the Laplacian of this convolution.

$$\boldsymbol{C}(\boldsymbol{x},\boldsymbol{y}) = \nabla^2\left(\boldsymbol{G}(\boldsymbol{r},\boldsymbol{\sigma}) \otimes \boldsymbol{I}(\boldsymbol{x},\boldsymbol{y})\right) \qquad \{3\text{-}1\}$$

For the sharp edge, the slope of the function ***C(x, y)*** at the point of zero crossing is equal to the maximum rate of change in image intensity, and hence can be used to estimate $\boldsymbol{\sigma}$ and from that, the distance to the imaged point. A limiting factor of this technique is that scene characteristics must be known prior to evaluation. Hence, a depth map produced by this means would be feature dependent, and offer very low spatial resolution over the image area. Pentland only gives qualitative results from the point of view of human perception.

Cardillo, et al [5] employed the same technique as Pentland [1], and used a 2-D Gaussian model to represent the psf. However, Cardillo's main interest was in using the image coordinates and focus blur to produce a calibration system capable of converting these parameters to absolute world coordinates. Never the less, he claims specific depth accuracies of 1.9% over a range of 750mm, though – of course – this is feature dependent.

3.2 Multiple-Image with Passive Illumination and Spatial Analysis Methods

A summary of featured research using this method is given in Table 3.2 below.

Ens & Lawrence [6] proposed a matrix-based method as a means of avoiding the problems caused by inverse filtering when used to obtain the defocus model. These problems include inaccuracies in determining frequency domain representation, windowing effects, and border effects. Their method is independent of any particular defocus model.

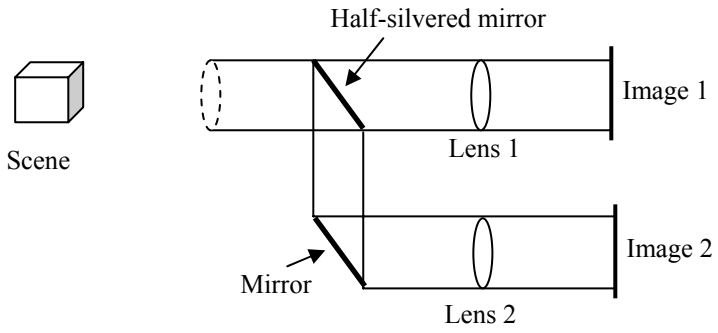


Figure 3.1 Simultaneous Image Capture

Two images are captured as shown in Fig 3.1. Each image has a different focus operator ***h(x, y)*** as they are acquired through apertures of different size. Their matrix-based method seeks to deconvolve the defocus operator from the images by characterising the problem as a system of linear equations (See Fig 3.2).

Previous Work Using MULTIPLE IMAGE with PASSIVE Illumination and SPATIAL Analysis Methods					
Author Reference	**Method**	**Assumed Defocus Operator**	**Spatial Resolution**	**Depth Range**	**Error**
Ens & Lawrence [6] 1991	Two image matrix-based – deconvolve the defocus operator from image by series of linear equations. Convolution matrix in lookup table	Experimentally derived	generally feature depend't	15mm @ distance of 1m	1.3% RMS
Surya & Subbarao [8]1993	S Transform, approximate the image function $f(x,y)$ as a cubic polynomial in small regions. Use lookup to convert σ to distance	Not specific		≈ 4m	2.3% @0.6m 20% @5m
Xu S, et al [10] 1995	Gradient of focus modelled from images taken at variable positions of image plane. Normalised power spectrum FSC used to determine exact focus. Depth estimated from arc of best focus. Microscopic.	NA	Related to change of image plane distance Δl	Very small Δl ≈±0.06 mm	~2Δl
Ziou D [15] 1997	The blur difference between two images can be computed by resolving a system of Hermite (orthogonal) polynomial equations.	Gaussian	?	~10cm @ distance of 2m	?
Rajagopalan A N and Chaudhuri S [18] 1999	The focused image of the scene and the space variant blur are modelled as MRF's. From two defocused images, estimation of blur is posed as a MAP problem.	Gaussian	?	~100cm	4%RMS
Rayala J,et al [22] 2001	Defocus is modelled as a linear system with an appropriate transfer function. Two images (focused & defocused) form input & output of system. Parametric transfer function is utilised to provide analytical expression for blur parameter	Generalised	Not specified	~30mm @ distance of 2m	Not specified
Favaro P and Soatto S [23,24, 27] 2002	A number of deblurred images are analysed via a set of linear operators (matrices) to produce a generalised model of the blur operator	Generalised	Not specified	Synthetic 330mm @ at distance of 0.5m	Synthetic 0.5%RMS
Simon C, et al [25] 2002	Uses two images – one sharp, one blurred – and determines ratio of gradient for points on thick edge as a means of estimating depth	Gaussian	Feature dependent	Only simulation	~1% to 3% simulated
Deschenes F, et al [26] 2004	A cooperative and simultaneous estimation of depth from defocus blur and spatial shift.	Gaussian	Not specified	<100mm	Only relative error specified

Table 3.2

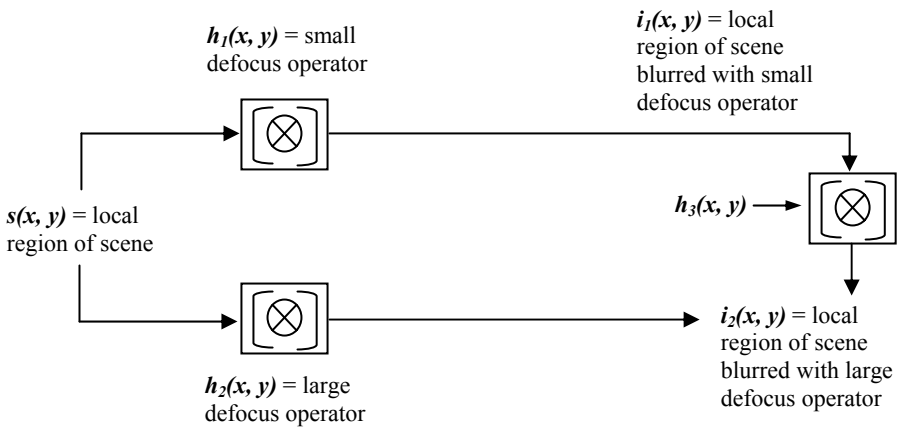


Figure 3.2 Block diagram of the Ens & Lawrence [6] approach to DfD

The Ens & Lawrence [6] method is described in Figure 3.2 where ***s(x, y)*** is a local region of the scene and ***h_1(x, y)*** is the **psf** caused by defocus through the smaller aperture. Hence, the acquired local region ***i_1(x, y)*** is determined from

$$\boldsymbol{i}_1(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{s}(\boldsymbol{x},\boldsymbol{y})[\otimes]\boldsymbol{h}_1(\boldsymbol{x},\boldsymbol{y}) \quad \{3\text{-}2\}$$

[3.1]

Similarly, ***h_2(x,y)*** is the **psf** caused by defocus through the larger aperture. Hence,

$$\boldsymbol{i}_2(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{s}(\boldsymbol{x},\boldsymbol{y})[\otimes]\boldsymbol{h}_2(\boldsymbol{x},\boldsymbol{y}) \quad \{3\text{-}3\}$$

The goal is to determine the function ***h_3(x, y)*** that will transform ***i_1(x, y)*** into ***i_2(x, y)*** according to the convolution,

$$\boldsymbol{i}_1(\boldsymbol{x},\boldsymbol{y})[\otimes]\boldsymbol{h}_3(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{i}_2(\boldsymbol{x},\boldsymbol{y}) \quad \{3\text{-}4\}$$

The distance from the camera to that local region is then uniquely related to ***h_3(x, y)***. ***h_3(x, y)*** is called the convolution ratio of the two defocus operators. Ens & Lawrence isolate ***h_3(x, y)*** by a matrix-based method whereby the convolution equation {3-4} can be written as a matrix equation,

$$[\boldsymbol{i}_{1BT}] \bullet \boldsymbol{h}_{3S} = \boldsymbol{i}_{2S} \quad \{3\text{-}5\}$$

[3.1] The operator [⊗] denotes restricted convolution where the borders of the kernel ***h_1(x,y)*** are not convolved beyond the borders of the scene ***s(x, y)***.

where $[i_{1BT}]$ is a N^2xN^2 block Toeplitz matrix constructed from $i_1(x, y)$, and both h_{3S} and i_{2S} are row-stacked vectors created from $h_3(x, y)$ and $i_2(x, y)$ respectively.

They show that though h_3 can be calculated using dimensions as small as $N = 3$, with the addition of noise solving {3-5} for h_{3S} becomes a very ill posed problem. However, on the assumption that h_3 must belong to a known family of patterns, they suggest a regularised[3.2] form of {3-5}

$$\left\| [i_{1BT}] \bullet h_{3S} - i_{2S} \right\|^2 + \lambda \left\| [C] \bullet h_{3S} \right\|^2 = \text{minimum} \qquad \{3\text{-}6\}$$

Solving {3-6} is computationally expensive, and [C] proves difficult to determine, so they use a generalised form of {3-4} and calculate *a priori* a table of $h_3(x, y)$ patterns, each corresponding to a known scene depth. Hence, for any given image regions $i_1(x, y)$ and $i_2(x, y)$ the table of $h_3(x, y)$ patterns can be searched iteratively for the best $h_3(x, y)$ that minimises,

$$\sum_{x=0}^{N-k} \sum_{y=0}^{N-k} \left[i_1(x, y) [\otimes] h_3(x, y) - i_2(x, y) \right]^2 = \text{minimum} \qquad \{3\text{-}7\}$$

where the size of $i_1(x, y)$ is $N \times N$, and the kernel $h_3(x, y)$ is $k \times k$. Other methods of recovering $h_3(x, y)$ are shown, and the author claims experimental error of 1.3% RMS error in depth estimation.

Surya & Subbarao [8] also used two images via different aperture settings, though both were blurred in this case. They used a (then) recently developed Spatial-Domain Convolution/Deconvolution Transform or S-Transform[3.3] where a rotationally symmetric psf, $h(x, y)$, is assumed. However, the method generates depth estimates with an error of ~2.3% at 0.6m ranging to ~20% at 5m. Consequently they conclude that this method could provide a rough distance of

[3.2] A.N.Tihonov, "Solution of incorrectly formulated problems and the regularisation method", *Soviet Mathematics, Translation of Doklady Akademii Nuak SSSR*, 4:1035-1038, 1963

[3.3] M.Subbarao, "Spatial-Domain Convolution/Deconvolution Transform", *Tech.Report No.91.07.03, Computer Vision Laboratory ,Dept of Elect Eng, SUNY, Stony Brook, NY,* 11794-2350

measurement that can then be used by a stereo algorithm to obtain a more accurate estimate of distance.

Xu S, et al [10] modelled the gradient of focus – rather than modelling the psf – over a number of images taken at various positions on the image plane. The centre of a curve *J(t)* indicates the position of exact focus. A normalised focus sharpness criteria (FSC) measuring the power spectrum within the 16 x 16 window was used to determine exact focus, the object distance l_o being the derivative of *J(t)*. Experimental data for this method was acquired over only a very small range of approximately 0.06mm.

Ziou D [15] computed the blur difference between two images by resolving a system of Hermite (orthoganol) polynomials[3.4]. An algorithm is presented for the estimation of blur in 1D and 2D cases, although results are rather qualitative.

Rajagopalan & Choudhuri [18] modelled the focused image of a scene and the space-variant blur parameter as Markov Random Fields (MRFs)[3.5]. From two defocused images of the scene, estimation of the blur is posed as a MAP problem (Maximum a Posteriori estimate). An MRF is a model of the joint probability distribution of a set of random variables (*X*), and can represent cyclic dependencies. Values in the random field have some form of spatial correlation with respect to the neighbourhood system, and are mapped onto a space of '*n*' dimensions. A posterior distribution is based on previous knowledge – in this case of typical neighbourhood structures – and the form of the MRF is derived from this. The method is computationally intensive and – given the level of complexity – the results are modest (4% RMS at a range of ~100cm). (See Figure 3.3 below).

[3.4] J.B.Martens, "The Hermite Transform – Theory", *IEEE Trans on Acous.Sig.&Speech Proc, 38(9):1595-1606*, 1990

[3.5] J. Subrahmonia, Y.P. Hung, D.B. Cooper, "Model-Based Segmentation and Estimation of 3-D Surfaces from Two or More Intensity Images Using Markov Random Fields", *Proc. IEEE Conf.Pattern Recognition pp.390-397*, 1990

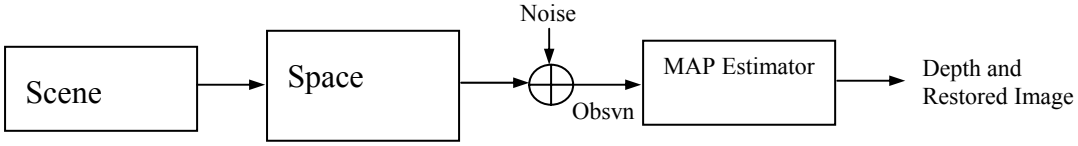


Figure 3.3 Schematic of Rajagopalan & Choudhuri [18] Method

Rayala, Guptal, & Mullick [22] also acquire two defocused images from different camera parameter settings, and use these to estimate the spread (σ) of the PSF by factoring out the contribution of the scene (i.e. they do not depend on 'fixed models' of PSF). Their method is to assume that a region of one defocused image ***x(s, t)*** will equal the same region of the second defocused image ***y(s, t)*** if it is blurred with an appropriate defocus operator ***h(s, t;*** σ_S***)***, hence

$$y(s,t) = x(s,t) * h(s,t;\sigma_S) \qquad \{3\text{-}8\}$$

As {3-8} represents a linear system with ***h(s,t;***σ_S***)*** as the impulse response, the transfer function of the system is seen to be,

$$H(\xi,\eta) = \frac{Y_C(\xi,\eta)}{X_C(\xi,\eta)} \qquad \{3\text{-}9\}$$

where ***H(*** ξ***,*** η***), Y*****C****(*** ξ***,*** η***),*** and ***X*****C****(*** ξ***,*** η***)*** are the Fourier Transforms of ***h(s,t;***σ_S***)***, ***y(s, t)***, and ***x(s, t)*** respectively. This transfer function ***H(*** ξ***,*** η***)*** is then approximated with a polynomial transfer function ***F*****d****(*** ξ***,*** η***)*** that represents a 2-D[3.6] sequence with ***b*****mn*** as coefficients. Hence,

$$F_d(\xi,\eta) = \sum_{m=-M}^{M} \sum_{n=-M}^{M} b_{mn} e^{-i\xi n\Delta t} e^{-\eta n\Delta t} \qquad \{3\text{-}10\}$$

An algorithm is developed to take local regions from the image pair, treat them as input and output to the system, and use a least-squares error technique to estimate the coefficients of the polynomial transfer function, and hence blur spread σ_S and depth.

[3.6] The Fourier Transform is a two-sided as the camera PSF is usually rotationally symmetric.

Actual results in terms of depth error are not specifically given, but the method appears complex, computationally expensive, and heavily reliant on adequate scene texture.

Favaro, et al [23, 24, 27] explored related approaches to recover shape from defocus. They proposed two methods. One (when the PSF is known) involves computing orthogonal operators from a number of defocused images of the scene. These operators are regularised via functional singular value decomposition[3.7]. The second method (when the form of the PSF is unknown) 'learns' a set of projection operators from blurred images and then uses these to estimate the 3D geometry of a scene from blurred images. The technique is effective in recovering the 3-D geometry of a scene over a limited depth range, but geometric form appears to be the main interest rather than specific depth mapping over a defined depth range.

Simon, et al [25] relied upon using a Prewitt operator to determine edge positions, and by acquiring two images (one sharp, one blurred) they generate a local depth estimation based on the ratio of edge gradient between corresponding locations in each image. For the PSF they assume a 2-D Gaussian model with a spread parameter σ_{So} corresponding to a depth s_o. The estimate of σ_{So} from points x_o (sharp) and $x_o+\varepsilon$ (blurred) are prone to noise, so each point belonging to the edge is used to create more estimates and reduce sensitivity to noise. Experimental trials were limited to synthetic images, and weaknesses of the method include (a) valid only for edge orientations of $\theta = 0 + n\frac{\pi}{2}$ (severely limits number of points for blur estimation), and (b) the influence of closely related neighbourhood points also causes error.

[3.7] Singular value decomposition (SVD) is a factorization of a rectangular matrix. In general, SVD[m] returns a list $\{U,D,V\}$ where U & V are matrices and D is a diagonal matrix composing the singular values of m.

Deschenes F, et al [26] presented a cooperative and simultaneous estimation of depth cues based on defocus blur and spatial shifts. They acquired two images of the same scene, but implemented the method through stereo disparity, 2-D motion, or zooming disparity. It is the results of zooming disparity that is of most relevance to this thesis, as that is more appropriate for comparison to the monocular techniques investigated. The algorithm is based on a generalised moment expansion where the blurred image is expressed as a function of the partial derivatives of the two images, the blur difference, and the horizontal and vertical shifts. These depth cues are determined by resolving a system of equations. Various values of relative error are quoted for numerous aspects of the process, and for differing implementations of the technique. They confine their results only to an evaluation of how successfully they are able to simultaneously compute defocus blurs and spatial shifts, rather than making any direct measurements of actual depths. Figure 3.4 indicates the type of results they presented in relation to this.

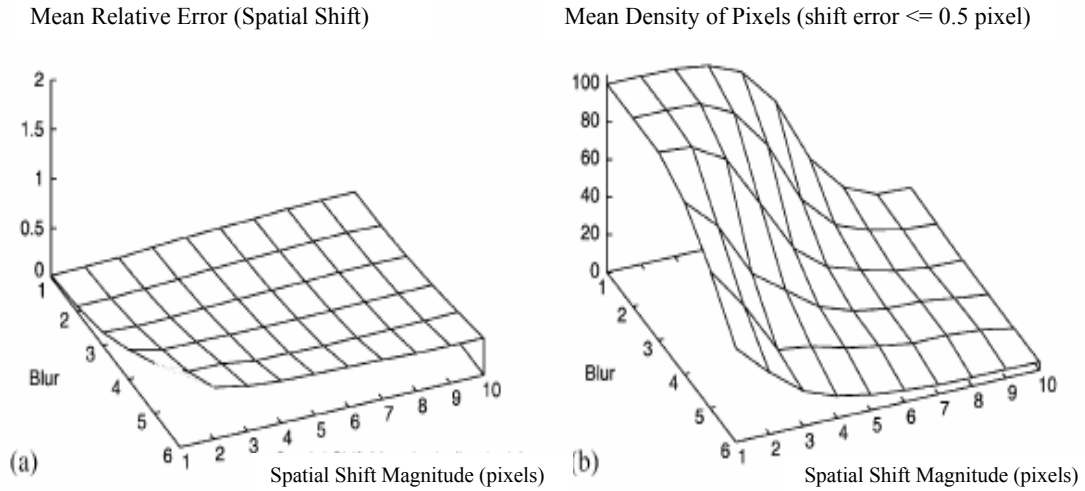


Figure 3.4 From Deschenes, et al [26]

(a & b) Mean estimation errors and densities of the spatial shift as a function of the defocus blur and the spatial shift magnitude.

3.3 Multiple-Image with Passive Illumination and Frequency Analysis Methods

Previous Work Using MULTIPLE IMAGE with PASSIVE Illumination and FREQUENCY Analysis Methods					
Author Reference	**Method**	**Assumed Defocus Operator**	**Spatial Resolution**	**Depth Range**	**Error**
Pentland [1,3]	Two images via beam-splitter – differing aperture hence DoF – compare corresponding Fourier Transform patches	Gaussian	128x128	1m	2.5%RMS
Subbarao [2]	Mathematical hypothesis – change blur parameter over two scene images by changing camera parameters. *Experimentally weak*	Gaussian	---	---	---
Subbarao & Wei [7]	One-dimensional Fourier coefficients, three images	NA	Per row	>50cm	6%
Pentland, et al [9]	Two images at different DoF via aperture. Window with Gaussian to reduce large blur effects. Compare windows using Fourier function	Gaussian	64x64	≈70 to 300cm	2.5% RMS
Rajagopalan, Chaudhuri [12]	Different types of Space Frequency Representation used to analyse two defocused images (changed camera parameter). Provides estimate of blur parameter, hence depth estimation.	Gaussian	- (32x32 window used)	10cm @12cm tested	~5%
Watanabe, Nayar [13]	A class of broadband operators – used together to provide invariance to scene texture, and accurate & dense depth maps.	Not specific	7x7 window	34cm@ 700cm	0.5 ~ 1.2%

Table 3.3

Pentland et al [1,3] measured the error in focus by comparing two geometrically identical images, one with a wide aperture where objects off the focal plane are blurred, and one with a small aperture where everything is sharply in focus. Depth estimation was made by analysing the 2-D Fourier transform of two corresponding image 'patches' ($f_1(r,\theta)$ and $f_2(r,\theta)$) to find the depth-related blur parameters σ_1 and σ_2 respectively. A Gaussian PSF, $G(r, \sigma)$ was assumed.

$$\frac{f_1(r,\theta)}{f_2(r,\theta)} = \frac{f_0(r,\theta) \otimes G(r,\sigma_1)}{f_0(r,\theta) \otimes G(r,\sigma_2)} \quad \{3\text{-}11\}$$

Pentland does not fully evaluate the technique for absolute depth measurement capability, but makes comparisons against subjective human impressions of depth.

[3.8] {3-11} can be substantially in error where large amounts of defocus exist due to the 'spread' of neighbouring patches

However, he does conclude that the aperture-algorithm provides much stronger information about the scene (compared to his earlier 'sharp edge' approach [1]), though the scene must contain both adequate image resolution and high-frequency content.

Subbarao [2] also investigated depth recovery by measuring the change in the scene's image due to a known change in a camera parameter. As with Pentland [1], Subbarao investigated changing the diameter of the lens aperture, but also investigated changing the lens to image detector distance, and changing the focal length of the lens. In fact he suggests that his passive range finding method can use any one – or a combination of – these three possible variations. Subbarao was able to change the focal length of the lens because he uses a two-lens arrangement (see Figure 3.5). This technique was discussed in section 2.3.3 of chapter 2.

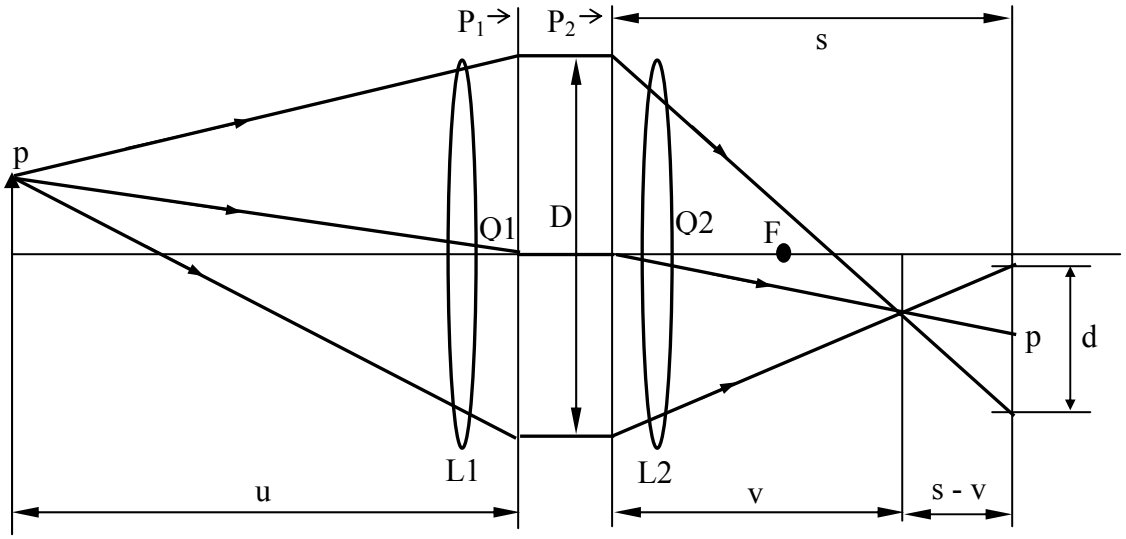


Figure 3.5 Two-lens Arrangement – Subbarao [2]

From Subbarao [2] - Showing the two-lens arrangement that allows the focal length of the lens to be a variable camera parameter. L_2 (and hence principal plane P_2) is moved relative to image detector plane I.

Subbarao [2] essentially uses the same premise that a depth estimate can be achieved by convolution of the sharp image with a PSF of appropriate blur parameter σ to produce the blurred image – the depth then being related to the calculated value of σ. However, he also contends that – in order to locate all corresponding points or regions in images at different depth values – the images should be scaled to have the same magnification. Gray-level rescaling is also used in order that mean gray level values are the same for all images. (*Later work e.g. [11], [13], [14] uses a Telecentric lens to avoid magnification effects when changing the image focus*).

To avoid the problem of neighbourhood overlap, Subarrao also suggests multiplication of the image intensity by a suitable centre-weighted (e.g. Gaussian) mask centred at the region of interest. Because the weights are higher at the centre and weaker at the periphery, neighbourhood effects are attenuated, and the depth estimate is concentrated to an approximation along the centre of the field of view.

A 2-D Gaussian PSF is assumed in the depth estimation method,

$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\frac{(x^2+y^2)}{\sigma^2}} \qquad \{3\text{-}12\}$$

where σ is the blur spread diameter, and the Fourier Transform of $h(x, y)$ is,

$$H(\omega,\upsilon) = e^{-\frac{1}{2}(\omega^2+\upsilon^2)\sigma^2} \qquad \{3\text{-}13\}$$

where ω, ν are spatial frequencies in radians per unit distance. Given the standard convolution equation relating the blurred image to the sharp image via a Gaussian PSF, i.e.

$$g(x,y) = h(x,y) * f(x,y) \qquad \{3\text{-}14\}$$

Hence, the power spectral density for a Gaussian PSF is,

$$P(\omega,\upsilon) = e^{-(\omega^2+\upsilon^2)\sigma^2} FF^* \quad {}^{3.9} \qquad \{3\text{-}15\}$$

[3.9] $F(\omega, \nu)$ is the Fourier transform of $f(x, y)$. * represents the convolution operation

From equation {3-15}, the ratio of power spectral densities for the two regions under consideration is,

$$\frac{P_1(\omega,\upsilon)}{P_2(\omega,\upsilon)} = e^{-(\omega^2+\upsilon^2)(\sigma_1^2-\sigma_2^2)} \qquad \{3\text{-}16\}$$

The difference in blur parameter between the regions of the two acquired images can be found be taking logarithms on either side of equation {3-16} and re-arranging the terms as shown in equation {3-17}.

$$\sigma_1^2 - \sigma_2^2 = \frac{-1}{\omega^2+\upsilon^2} \ln\left[\frac{P_1(\omega,\upsilon)}{P_2(\omega,\upsilon)}\right] \qquad \{3\text{-}17\}$$

Subbarao also gives a variation on this method for the purpose of estimating depth recovery from small changes in camera parameter. However, experimental results are very limited, and little indication of actual depth accuracy is given. Rather, a number of difficulties are pointed out, including the fact that depth estimation would be more robust if more images are used, and that the actual form of the camera point spread function could be significantly different from the Gaussian model used. Other problems include estimating values of the camera parameters, illumination condition of the scene, aberrations of the optical system, image quality (spatial and grey-level resolution), and the inability of the system to cope with texture-less surfaces, etc. At one point this leads him to propose that the method could provide such qualitative information as "object A is nearer than object B", or "there are no obstacles within distance X". This seems to suggest a very un-ambitious result for a relatively complex process.

In a later work, Subbarao & Wei [7] proposed a multi-image method based on computing one-dimensional Fourier coefficients. They contend that their system can determine the distance of an object from 0.5 m to infinity, but give the error – not of the estimated distance – but of the accuracy of the lens focus position (6% RMS).

Using a very similar approach Pentland et al [9] also acquired two images at different depth-of-field via aperture variation. Corresponding local region windows were compared using a 2-D Fourier transform so that the original irradiance distribution is factored out. Assuming a 2-D Gaussian PSF model they argued that the distance to the image point is a monotonically decreasing function of $\ln I_1(\omega_r) - \ln I_2(\omega_r)$[3.10], the difference in localised Fourier domain power between the two cameras at corresponding image locations. Their working range was 70 cm to 300 cm, and error of 2.5% RMS was claimed for this passive method.

Rajagopalan, Chaudhuri [12] analyse two acquired images using different forms of space frequency representation (SFR). This provides estimates of the blur parameter, and by assuming a Gaussian PSF, estimates of depth are derived. However, estimates are noisy and have a large variation over a neighbourhood. They adopt a variational approach where a smoothness constraint is applied to the estimates. Their method was only tested over a range of ~10 cm, with declared error of ~5%.

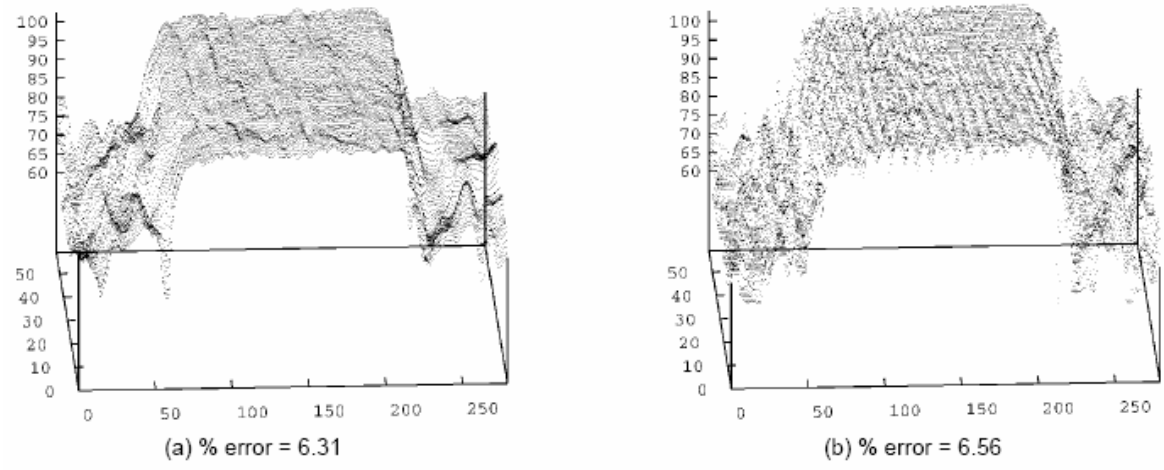


Figure 3.6 Estimates of depth using various SFR methods – Rajagopalan, Chaudhuri [12]

[3.10] ω_r is the radial frequency of the window region.

Watanabe & Nayar [13] proposed a set of broadband operators that can model relative image blur in the frequency domain. When used together, these operators provide invariance to scene texture, and are claimed to produce accurate and dense depth maps. The operators are derived using a rational expression to model the relative image blur. They can be derived – in theory – for any image blur model, and hence are not dependent on assumed standard PSF models. A 7x7 size operator was used to give relatively high spatial resolution. Experiments covered a depth range of 34 cm at a distance of 700 cm, with error (increasing with distance) at 0.5% to 1.2%. These results are very good results compared to many methods discussed.

3.4 Single-Image with Active Illumination Methods

Previous Work Using SINGLE IMAGE with ACTIVE Illumination Methods						
Author Reference	**Method**	**Analysis Domain**	**Assumed Defocus Operator**	**Resol'n**	**Depth Range**	**Error**
Girod & Scherock [4]	Measure spread of projected stripe widths – anisotropic aperture projection – astigmatic lens (2 planes of focus, v & h) *Poor spatial resolution*	Spatial	---	23x hor	---	---
Hinojosa C, et al [17]	Project a horizontal fringe pattern using laser and diffraction grating,. Measure width of fringes and relate to a Gaussian spread function.	Spatial	Gaussian	20x100 2000 depth estimates	?	Not stated.

Table 3.4

Girod & Scherock [4] produced one of the early pieces of work using active structured projected light as a means of gaining independence from the constraints of object surface texture and ambient light conditions. Their method projected a pattern of periodic lines along the optical axis. The line spread due to defocus was measured using,

$$\sigma_{stripe} = \sqrt{\frac{\sum_x x^2 L(x) - \left(\sum_x xL(x)\right)^2}{\sum_x L(x)}} \qquad \{3\text{-}18\}$$

where ***x*** is the column number relative to the centre of the line, and ***L(x)*** is the luminance due to the projected light. The spatial resolution of the generated depth map is severely limited due to the pattern used (23 lines per image), and aliasing effects produced very ragged edges. They proposed that defocus effects due to camera parameters (depth of field, etc) are approximately isotropic, and that error due to this effect can be offset by projecting an isotropic pattern via a shaped aperture. They further propose that projection of the structured pattern could be through an anisotropic lens system with astigmatic optics. This would effectively provide planes of best focus for horizontal or vertical light patterns allowing depth to be recovered in these two orthogonal directions.

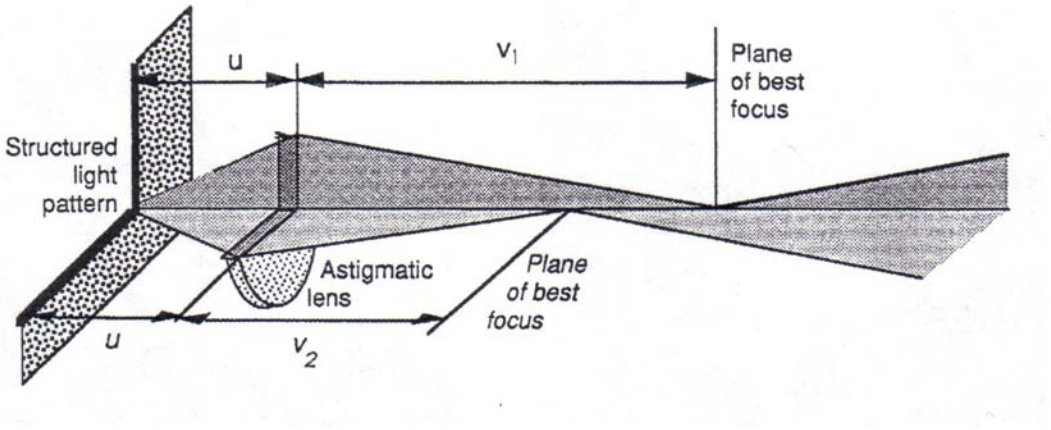


Figure 3.7 A DfD light source with astigmatic optics - Girod & Scherock [4]

Hinojosa C, et al [17] also estimated depth using defocused structured light (DSL) and only requiring a single image. They use a laser light source (see Figure 3.8) expanded by a spatial filter, a 1-D grating, and a series of spherical and cylindrical lenses to create the pattern of lines (fringes). The image is captured by an off-axis camera. Their depth recovery method determines the central position of each DSL fringe, and then estimates the width of the blurred line σ using a least squares error estimation in relation to a Gaussian spread function.

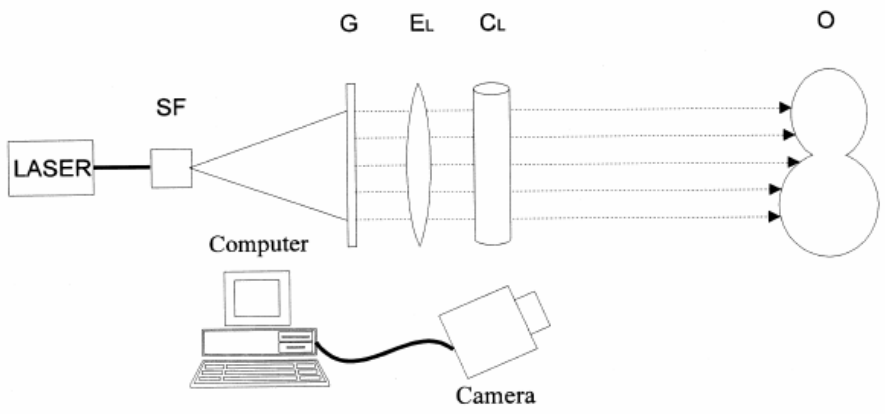


Figure 3.8 Fringe pattern projection system – Hinojosa C, et al [17]

Hinojosa does not give direct depth estimate errors, but the depth map appears to show a resolution of approximately 20 x 100 estimates per image.

3.5 Multiple-Image with Active Illumination Methods

Pentland, et al [9] projects a pattern of parallel lines of single pixel width along the optical axis. A smoothing algorithm is applied to the acquired image in order to reduce noise and smooth intensity 'humps'. They measure the moment of inertia for each intensity 'hump' (I_y) corresponding to a pattern line. A constant of proportionality is determined empirically for each value of I_y by using a calibrated ramp. These values are then entered in a lookup table. Hence, the lookup table relates I_y to an estimated depth. Experimentally, they worked on a range of 5 cm and claimed depth estimate error of 0.5% RMS. Resolution was limited to pattern periodicity.

Nayar, et al [11] set out to produce an active illumination real time focus range sensor, where all aspects of the DfD system were configured to produce optimum performance. They used a beam splitter to acquire two simultaneous images of the same scene (see Figure 3.9), but with the two images being focused at different positions along the optical axis.

Previous Work Using MULTIPLE IMAGE with ACTIVE Illumination Methods						
Author Reference	**Method**	**Spatial or Frequency Analysis**	**Assumed Defocus Operator**	**Spatial Resolution**	**Depth Range**	**Error**
Pentland, et al [9]	Project parallel lines of single pixel width. Measure moment I_y of inertia of 'humps'. Use calibrated ramp to produce LUT of I_y v depth	Spatial	NA	pattern	5cm	0.5% RMS
Nayar, et al [11]	Optimised projected light pattern. Two images (via Telecentric optics) at different positions of focus. Relative blurring is analysed via narrow-band linear operator. Look up table maps each pair of focus measures to unique depth estimate.	Spatial though system elements are optimised via frequency analysis	NA	Spatial 256 x 240 simultan's grab 512 x 480 successive grab	30cm	0.24% simultaneous 0.34% successive
Kowarschik_R, Kuhmstedt_P, et al [20]	Object is successfully illuminated with fringe projections from different directions. Gray code is combined with 90deg phase shifts. At least 3 linearly independent phase measurements are measured by gray-code to calculate 3D coordinates	Spatial	NA	Not specific	0.5 to 70 mm	? 0.5 to 15µm measurement uncertainty
Ghita O Whelan P F [21]	Bifocal CCD system captures two simultaneous images with different focal settings. Projected structured light illuminates the object to provide dominant texture at known spatial frequency. Blur parameter (hence depth estimation) is recovered by filtering and relating focus differences between images	Spatial	Gaussian	Not specified	86cm	3.4%
Zhang L, Nayar S [28]	An illumination pattern with a wide range of frequencies is shifted across the scene. The surface radiance of a point over time is then the response of its defocus kernel to pattern illumination. Blur (& hence depth) is quantified by decomposing a discrete-time Fourier series and noting how the coefficients diminish.	Frequency	Determined through process	Per pixel claimed, but robust at discontinuities.	~60cm	Appears to be mainly qualitative

Table 3.5

They avoid zooming effects between the images by developing a Telecentric lens arrangement that is achieved by using an additional aperture fitted at the front focal point of the lens.

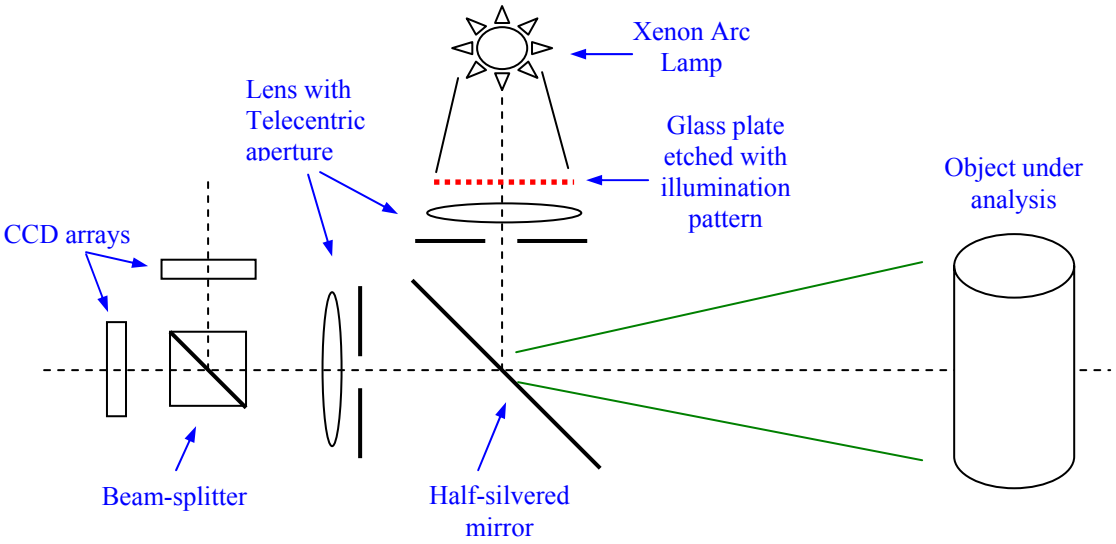


Figure 3.9 DfD system developed by Nayar et al [11]

The structured light pattern is projected along the optical axis via a half-mirror. As the pattern is projected through a lens and Telecentric aperture arrangement that is identical to that used by the camera system, the pattern retains the same perspective as that of the camera system that is viewing it.

The focal lengths of the two CCD sensors are offset by 0.25mm to give a working range of $\beta \approx 30$ cm between the points of focus (I_1 and I_2, see Figure 3.10) for each CCD sensor.

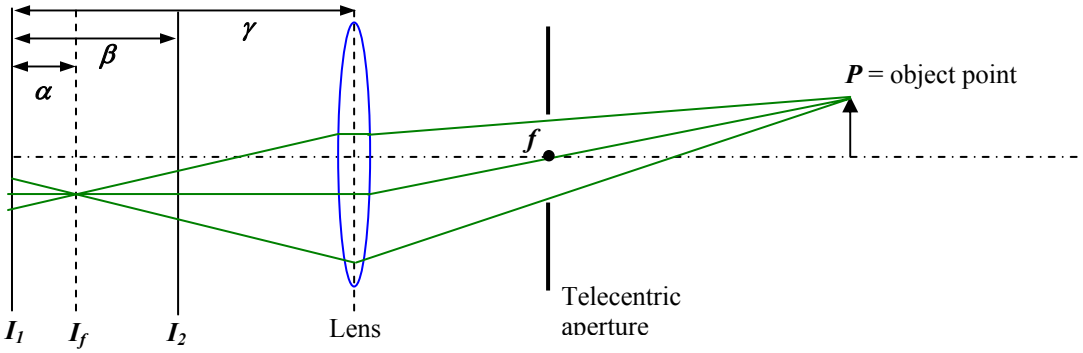


Figure 3.10 Two-image constant magnification imaging system. Nayar et al [11]

In order to achieve the optimum system design Nayar models five key components, the illumination pattern, the optical transfer function, the defocus model, and the focus operator. The depth of each scene point is determined by estimating the distance α (see Figure 3.10), and a tuned focus operator is applied to both images to obtain focus measure images $g_1(x, y)$ and $g_2(x, y)$. From this,

$$\frac{g_1(x,y)}{g_2(x,y)} = \frac{H\left[\frac{1}{t_x}, \frac{1}{t_y}; \alpha\right]}{H\left[\frac{1}{t_x}, \frac{1}{t_y}; \alpha - \beta\right]} \qquad \{3\text{-}19\}$$

where H is the defocus function in Fourier domain, and t_x, and t_y represent the period of the illumination pattern. The ratio formed in {3.19} removes pattern dependence from the estimation. A monotonic function $q = \frac{g_1 - g_2}{g_1 + g_2}$ is computed and stored in a look-up table that maps each image point to a unique value of α. This can be used in conjunction with the lens law to derive an estimate of depth. Nayar claims experimental error of 0.24% when the two images are 'grabbed' simultaneously, and 0.34% when the two images are acquired in successive 'grabs'.

Kowarschik_R, Kuhmstedt_P, et al [20] successively illuminated the object with grid type patterns from at least three different directions. Phase measurement values are determined in connection with simultaneous variation of intensity of the grid pattern.

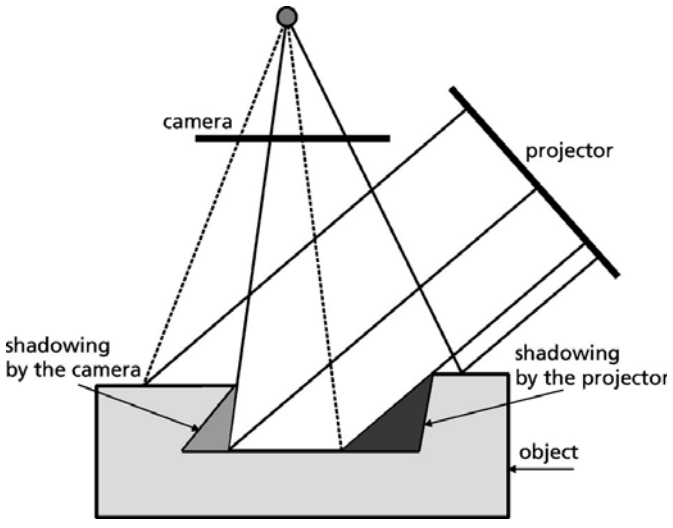


Figure 3.11 Shadowing by both camera and projector. (Kowarschik [20])

This minimises both shadowing effects and specular-type reflections. The object is further rotated on a second axis in order to produce a series of views that can be merged into a coordinate measurement system.

Ghita O and Whelan P F [21] used a two-image system similar to that of Nayar [11]. They illuminated the scene with a periodic line pattern, and filtered both the near- and far-focused images with a 5x5 Laplacian operator, the output of which indicates the level of focus. By assuming a Gaussian PSF, a depth estimate can be made.

Zhang L, Nayar S [28] projected an illumination pattern with a wide range of frequencies and shifted it across the scene. The surface radiance of a point over the time is then the response of its defocus kernel to pattern illumination. Blur – and hence depth – is quantified by decomposing a discrete-time Fourier series and noting how the coefficients diminish. Whilst this can give pixel-level resolution, their results appear relatively qualitative with no actual depth error details quoted.

3.6 Other Optical Methods of Estimating Depth

3.6.1 Neural Networks

A number of depth estimating systems based either entirely or partially on Artificial Neural Networks (ANNs) have been developed. Pham and Bayro-Corrochano [29] developed a two-stage process using a three-layer backpropagation multilayer perceptron that was trained on data clusters provided by the outputs of a Kohonen self-organising algorithm. This clustering technique has the advantage of being able to organise itself and produce outputs that indicate the degrees of membership of the different clusters (i.e. fuzzy outputs).

Mills H, Burton DR, Lalor MJ [30] presented work on using ANNs to analyse the intensity variations of fringe patterns. They compared the performance of

backpropagation networks, radial basis function networks, and fuzzy-artmap, but were more interested in their respective abilities to classify or interpolate rather than estimate depth directly. Cuevas FJ, Servin M, Rodriguez-Vera R [31] also analysed fringe pattern projections using radial basis functions. They analysed phase (related to object height) and used phase information from known calibration planes to train their ANN system. A *relative average error* of 2.6% compared to a contact measurement method was claimed.

Ma and Staunton [36] used a two-image method where the data was pre-processed via a multi-resolution pyramid that produced the training data required for the three-layered backpropagation neural network that generated the depth model.

Other similar ANN work involving multi-layer perceptron configurations can be found in [32], [33], [34], and for radial basis function networks in [35].

3.6.2 Photogrammetry

Photogrammetry defines a range of optical measurement techniques that typically uses multi-camera view-points of a scene that contains either coded targets or some other means of identifying common feature points in the multiple views acquired. Triangulation techniques can then be used to determine 3-D coordinates of those features. Although photogrammetry has a very wide range of applications including topographic mapping, building architecture, geology, archaeology, etc., examples of its use in modern engineering coordinate measurement can be seen in references [37 - 44].

3.7 Summary of Chapter 3

This chapter has summarised a range of influential research publications in the field of depth from defocus. Due to the extensive range of methods adopted by researchers, the chapter has been organised into sections, each of which represents research using a particular type of approach. The research reviewed in each section was organised broadly by whether the method used a single- or double-image technique, whether the illumination used was passive or active, and whether the method of analysing the depth information was in the spatial domain or the frequency domain.

Each chapter section also presented a table of summarised data for each piece of research work reviewed in that section. The intention of the table was to give a direct comparison of how well each technique performed when applied to actual depth measurements. Unfortunately, many researchers do not apply their reported technique to actual depth measurements, and so do not provide conveniently comparable results. They often simply conclude by reporting on the success of some other related aspect of their particular research.

In relation to the work presented in this thesis, one thing is clear from this review of the previous research work. This is the fact that the systems that have provided best results in terms of depth estimate accuracy to date have been systems based on a two-image technique.

Of the reported single-image systems, only Cardillo et al [5] reports actual depth estimate error of 1.9%, but with very low spatial resolution that is feature dependent. For the two-image systems some of the best results are claimed by Watanabe & Nayar [13] 0.5% to 1.2 % (passive illumination), and using active illumination Pentland et al [9] 0.5% RMS, Nayar et al [11] 0.24%.

Chapter 3 - References

[1] Pentland A P, "A New Sense for Depth of Field", *IEEE Trans. Pattern Analysis and Machine Intelligence*, v9, *No.4, pp.523-531*, July 1987.

[2] Subbarao M, "Parallel Depth Recovery by changing Camera Parameters", *Proc.Int.Conf.on Computer Vision*; *Tarpon Springs FL, IEEE; pp.149-155,* 1988

[3] Pentland A, et al, "A Simple Real-Time Range Camera", *Computer Vision and Pattern Recognition, Proceedings CVPR '89, IEEE Computer Society Conference on 4-8 June pp256 - 261*, 1989

[4] Girod B, Scherock S, "Depth from Defocus of Structured Light", *SPIE, Optics, Illumination, and Image Sensing for Machine Vision, v1194, pp209-215*; 1989

[5] Cardillo J, et al, "3-D Position Sensing Using a Single-Camera Approach", *Midwest Symposium on Circuits and Systems*, *pp. 325-328,* 1990

[6] Ens J and Lawrence P, "Investigation of methods for determining depth from focus ", *IEEE Transactions on Pattern Analysis and Machine Intelligence* v*15 No2, pp. 97-108,* 1991

[7] Subbarao, Murali, Wei, Tse-Chung, Surya, Gopal, "Focused image recovery from two defocused images recorded with different camera settings ", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *pp. 786-791*, 1994

[8] Surya G, Subbarao M, "Depth from Defocus by Changing Camera Aperture: A Spatial Domain Approach", *IEEE Computer Vision and Pattern Recognition*, *pp. 61-67*, 1993

[9] Pentland A P, Scherock S, Darrell T, Girod B, "Simple range cameras based on focal error", *J.Opt.Soc.Am; v11 No11; pp.2925-2934*, 1994

[10] Xu S, et al, "Range measurement from defocus gradient", *Machine Vision and Applications; vl8; pp.179-186*, 1995

[11] Nayar SK, Watanabe M, Noguchi M, "Real-time Focus Range Sensor", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *v18 No.12, pp1186-1197,* 1996

[12] Rajagopalan A N and Chaudhuri S, "A Variational Approach to Recovering Depth From Defocused Images", *IEEE PAMI; v19, No10, pp1158-1164*; 1996

[13] Watanabe M and Nayar S K, "Minimal Operator set for Passive Depth From Defocus", *Proc.IEEE Comput. Soc. Conf. on Comput. Vision and Pattern Recognition San Francisco CA USA June; p431-438,* 1996

[14] Watanabe M and Nayar S K, "Telecentric Optics for Focus Analysis", *IEEE Trans PAMI; v19, No12, p1360-1365*, 1997

[15] Zui D, "Passive Depth from Defocus Using a Spatial Domain Approach", *Proceedings of the IEEE International Conference on Computer Vision*, *pp. 799-804*, 1998

[16] Tsai DM, Lin CT, "A Moment-preserving Approach for Depth from DeFocus", *Pattern Recognition, v31 No5, pp551-60*, 1998

[17] Hinojosa C, Serrano-Heredia A, Ibarra J, "Recovery of three-dimensional shapes by using defocused structured light", *Optics & Laser Technology; v30; p281-290*, 1998

[18] Rajagopalan A N and Chaudhuri S, "An MRF Model-based Approach to Simultaneous Recovery of Depth And Restoration", *IEEE PAMI; v21, No7, p577-589*, 1999

[19] Mair C and Goodman C J, "Diffraction Limited Depth From Defocus", *Electron Lett; v36, No24, pp2012-2013*, 2000

[20] Kowarschik_R, Kuhmstedt_P, et al, "Adaptive optical three-dimensional measurement with structured light", *OPTICAL ENGINEERING, v39, No1, pp.150-58*, 2000

[21] Ghita O Whelan P F, "A Video-rate Range Sensor Based on Depth From Defocus", *Opt Laser Technol; v33, No3, pp167-176*, 2001

[22] Rayala J Gupta S and Mullick S K, "Estimation of Depth From Defocus as Polynomial System Identification", *IEE Proc vis Image Signal Process, v148, No5, p356-362*, 2001

[23] Favaro P and Soatto S, "Learning Shape From Defocus", *Lecture Notes in Computer Science; A. Heyden et al; ECCV; v2351, pp735-745 (Springer-Verlag Berlin Heidelberg)*, 2002

[24] Jin H L and Favaro P, "A Variational Approach to Shape From Defocus", *Lecture Notes in Computer Science; A. Heyden et al; ECCV; v2351, pp18-30 (Springer-Verlag Berlin Heidelberg)*, 2002

[25] Simon C, Bicking F, Simon T, "Estimation of depth on thick edges from sharp and blurred images", *Conference Record - IEEE Instrumentation and Measurement Technology Conference; v1, pp323-328*; 2002

[26] Deschenes F, Ziou D, Fuchs P, "A unified approach for a simultaneous and cooperative estimation of defocus blur and spatial shifts", *Image and Vision Computing; v22, pp35-37*, 2004

[27] Favaro P and Soatto S, "A geometric approach to shape from defocus", *IEEE Trans PAMI, v27, No3, pp406-417*, 2005

[28] Zhang L, Nayar S, "Projection Defocus Analysis for Scene Capture and Image Display", *ACM Transactions on Graphics; v25, No3, pp. 907-915*, 2006

[29] Pham DT; Bayro-Corrochano EJ, "Self-Organising Neural-Network-Based Pattern Clustering Method with Fuzzy Outputs", *Pattern recognition, v27, No.8, pp1103-10*, 1994

[30] Mills H, Burton DR, Lalor MJ, "Applying Backpropagation Neural Networks to Fringe Analysis", *Optics and Lasers in Engineering; v23, pp331-341*, 1995

[31] Cuevas FJ, Servin M, Rodriguez-Vera R, "Depth object recovery using radial basis functions", *Optics Communications; v163 issue4-6, pp270-277*, 1999

[32] Pham DT, Aslantas V, "Depth from defocusing using a neural network", *Pattern recognition, v32, pp715-727*, 1999

[33] Cuevas FJ, et al, “Multi-layer neural network applied to phase and depth recovery from fringe patterns”, *Optics Communications; v181 No4, pp239-259*, 2000

[34] Asif M, Choi TS, “Shape From Focus Using Multilayer Feedforward Neural Networks”, *IEEE Transactions on Image Processing; v10, No11, pp1670-1675*, 2001

[35] Jong, S.-M.; Huang, J.-S., “Using radial basis function networks to approach the depth from defocus”, *Journal of Imaging Science and Technology; v45, No4, pp400-406,* 2001

[36] Li Ma, R.C. Staunton, “Integration of multiresolution image segmentation and neural networks for object depth recovery”, *Journal of the Pattern Recognition Society; v38, No7, pp985-996,* 2005

[37] Chang, L.-., Zhao, B.-., Yang, J.-. & Chen, L.-., "Optical system design of space three-dimensional photographic", *Guangzi Xuebao/Acta Photonica Sinica, v36, No3, pp539-542*., 2007

[38] Edmundson, K.L., “Beyond inspection: The promise of videometrics for industry and engineering.”; *Proceedings of SPIE - The International Society for Optical Engineering; v6491, No 64910Q*; 2007

[39] Ergün, B., "An expert measurement system for photogrammetric industrial application"**,** *Measurement: Journal of the International Measurement Confederation, v39, No5, pp415-419*. 2006

[40] Gardel Kurka, P.R. & Rudek, M., "Three-dimensional volume and position recovering using a virtual reference box"**,** *IEEE Transactions on Image Processing, v16, No2, pp573-576*. 2007

[41] Genovese, K., Cosola, E. & Pappalettere, C., “360-deg measurement on tubular samples with axial stereogrammetry.”; *Proceedings of the 2006 SEM Annual Conference and Exposition on Experimental and Applied Mechanics; v* 3, *pp1172-1181*; 2006

[42] Genovese, K. & Pappalettere, C., "Axial stereo-photogrammetry for 360° measurement on tubular samples", *Optics and Lasers in Engineering, v45, No5, pp637-650*, 2007

[43] Teutsch, C., Berndt, D., Sobotta, A. & Sperling, S., “A flexible photogrammetric stereo vision system for capturing the 3D shape of extruded profiles.”; *Proceedings of SPIE - The International Society for Optical Engineering; v6382, No63820M*, 2006

[44] Zhang, Y., Zhang, Z. & Zhang, J., "Automatic measurement of industrial sheetmetal parts with CAD data and non-metric image sequence"**,** *Computer Vision and Image Understanding, v102, No1, pp52-59*, 2006

Chapter 4

Investigating Depth Cues in a Projected Structured Light Pattern

This chapter provides details of experiments that were carried out in order to analyse the possible defocus effects that may be present in a projected structured light pattern, and to determine their effectiveness as a depth cue capable of relating defocus to depth. The projection pattern used was a pattern of black/white vertical bars, similar to that used by Girod and Scherock [1]. Depth cues that were investigated against a change in depth included,

- The width (in pixels) of the white bar (white bar in image centre was sampled)*
- The mark/space ratio of white bar to black bar*
- The standard deviation (in pixels) of a white bar between local intensity minima.
- The intensity variation (grey scale) of the pattern image.
- The period of pattern frequency (pixels).
- Variation in shape of the edge profile (black-to-white transition of the pattern edge).

*These variables were measured at an intensity level of 15% of that between local intensity minima (nominally centre of black bar), and local intensity maxima (nominally centre of white bar). This sampling level is highlighted in Figure 4.3. The reason for this is that the pattern edge profiles will tend to distort due to defocus *about the point of average intensity*. Hence, minimum variation in, e.g. mark/space ratio, would be detected if these variations were sampled at average intensity levels.

4.1 Initial depth cue experiments

Initial experiments were prompted by the work of Girod and Scherock [1] who projected a structured light pattern through a large aperture lens onto the object surface. By using a beam splitting mirror, the structured light pattern (essentially alternate black and white bars) was projected along the optical axis. The basis of their technique was to measure the spread (i.e. width) of the white bars as a function of defocus, and to relate this to a variation in depth from the point of best focus with respect to object position. Their technique was simple, and fast, but gave very poor resolution (only 23 depth estimates per image row).

Although later work (see [2] to [7]) using projected structured light produced better results in terms of depth accuracy and spatial resolution, they were computationally very much more complex. Hence the initial experiments described in this chapter attempted to re-examine the depth cues available from a projected structured light pattern, and to determine whether improved algorithms could be designed to retain the computational simplicity whilst producing high quality results.

It was possible to carry out some initial experiments into projected light pattern depth cues – and to begin the first aspects of system design – whilst waiting to acquire both optical bench equipment and a suitable high-resolution camera. A basic rig was assembled using a digital camera (Kyocera M410R 4.0 Mega Pixel), a data projector (Plus U2 1110), and a laptop computer to generate the structured light pattern (see Fig 4.1). Both camera and projector were focused on the white planar screen at a distance of 170 cm from the camera lens.

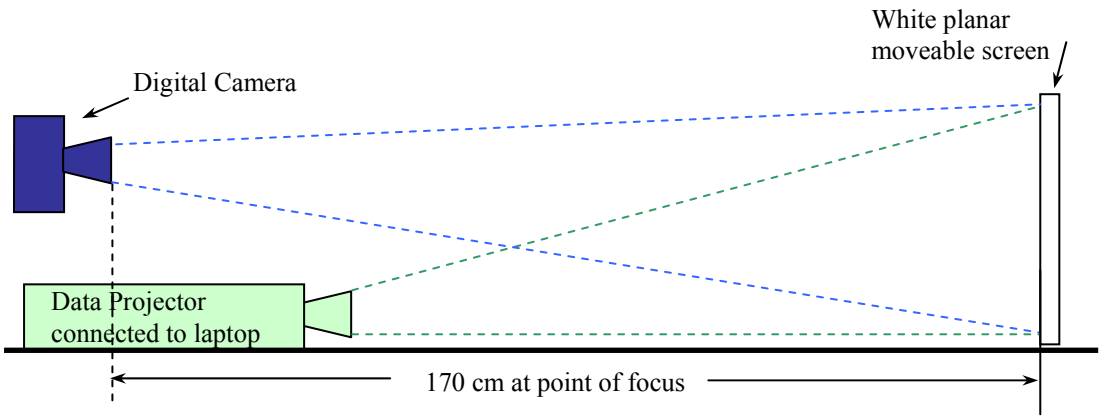
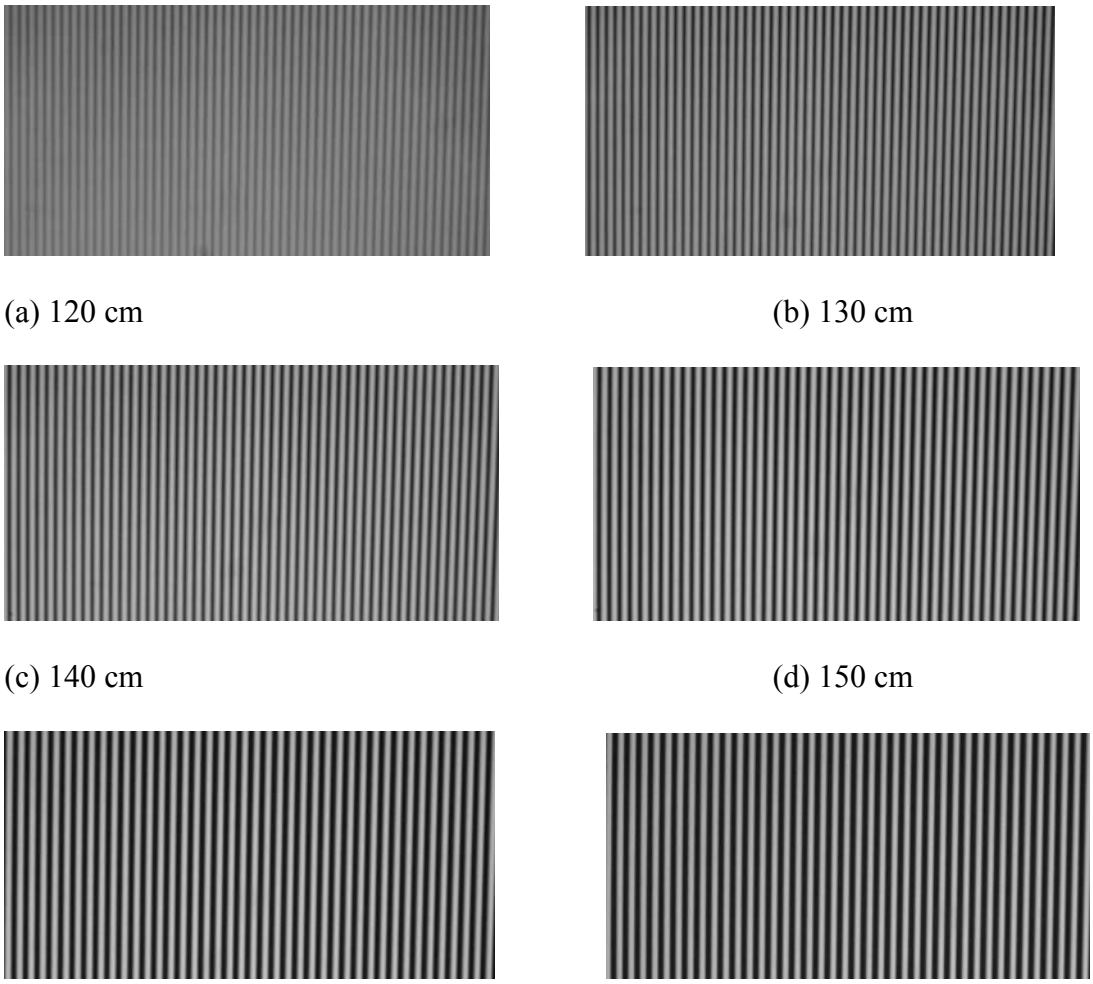


Figure 4.1 Preliminary Experimental Rig

The camera was zoomed to set the entire planar screen image within its field of view. A MATLAB routine (see Appendix A-1) was written to generate a projected black/white vertical bar pattern. As the planar screen was moved, an image of the projected light pattern on the screen was captured at each 10 cm variation in depth between the distances of 120 cm to 170 cm inclusive.

It should be noted that the digital camera was allowed to auto-focus at each position of the image screen, as the defocus of the projected pattern on the screen was the only defocus effect that was required for the purpose of this experiment.

This experimental rig had its limitations, but it was considered adequate for the initial investigation of depth cues and their variation over a restricted depth range. Six images were captured, and software was written to analyse the images in order to determine the variation in the effective width of the white bars due to depth from defocus. The six images can be seen in Figure 4.2 below.



(a) 120 cm (b) 130 cm

(c) 140 cm (d) 150 cm

(e) 160 cm (f) 170 cm

Figure 4.2 Images showing spread of projected vertical white bars due to defocus

The point of focus of the projected pattern was determined (by eye) to be at a distance of approximately 170 cm. Figure 4.2 (f) shows that, at this distance, the width of the vertical white bars are at their smallest in relation to the black bars, whereas as the projector to object plane distance decreases towards 120 cm the defocus effect causes the width of the white bars to increase. It can also be noted that an additional effect is a reduction in light intensity level due to defocus of the white bars of light.

At this point an image-processing algorithm was required in order to make specific analysis of the potential depth cues in the series of pattern-defocused images.

4.2 Development of a pattern analysis algorithm

Figure 4.3 illustrates the basis of the algorithm developed in the program 'VertStripeCalc.m' (See Appendix A-2). Figure 4.3 represents a section of the plot

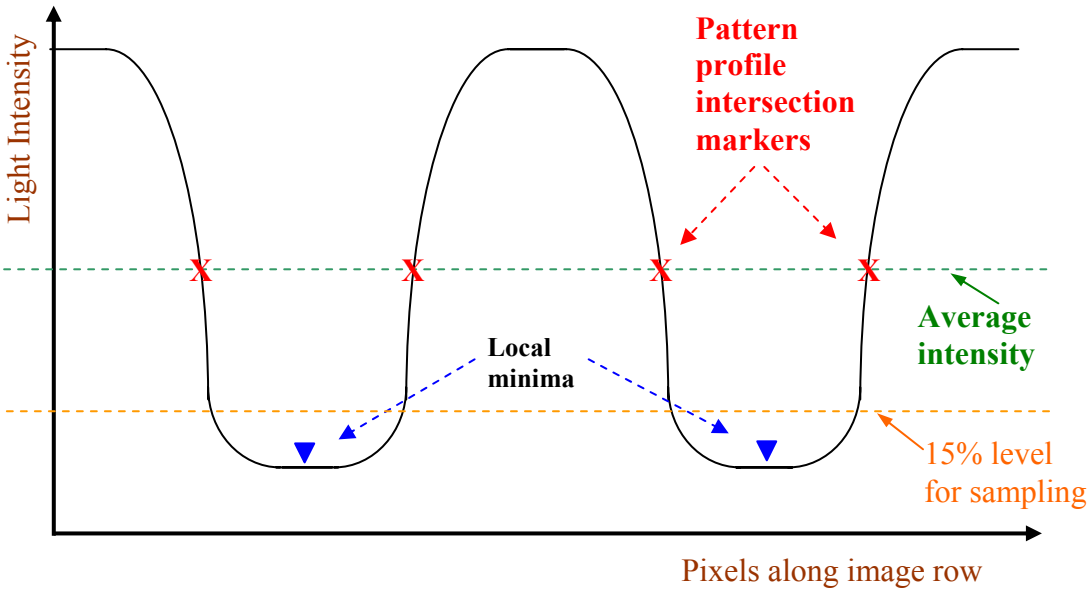


Figure 4.3 Illustration of the algorithm used in 'VertStripeCalc.m' to determine parameters to be used for comparison of structured light pattern depth cues

of the profile of the structured light pattern intensity level along a single row of the image. The first iteration of the program seeks the 'white bar' nearest the centre of the row, determines the average light intensity along the row, then uses the average value as a means of determing intersection markers as shown in the diagram. This gives a means of measuring a number of profile parameters that may be used as a metric to compare variations of defocus between each image at different depth. These could include,

- Mark/Space ratio of white bar to dark gap (at 15% sampling level)
- Width of 'white bar' (at 15% sampling level)
- Standard deviation of the width of the central white bar (between local mimima)

This algorithm formed the basis of more complex programs that would eventually be developed to analyse such plots using more sophisticated methods.

4.3 Experiment 1: Depth cue variation versus depth change

The objective of this first experiment was simply to determine whether a number of possible depth cues, i.e.

a. The nominal width of the projected white bar

b. The Standard Deviation of the pattern (between local minima, see Fig 4.3)

c. The pattern intensity variation

d. The mark/space ratio of the black/white pattern

would vary approximately linearly with defocus in relation to depth variation.

A version of 'VertStripeCalc.m' (App A-2) was first used to investigate the variation in the first two depth cues (a & b) listed above. A plot of the results is given in Figure 4.4 below.

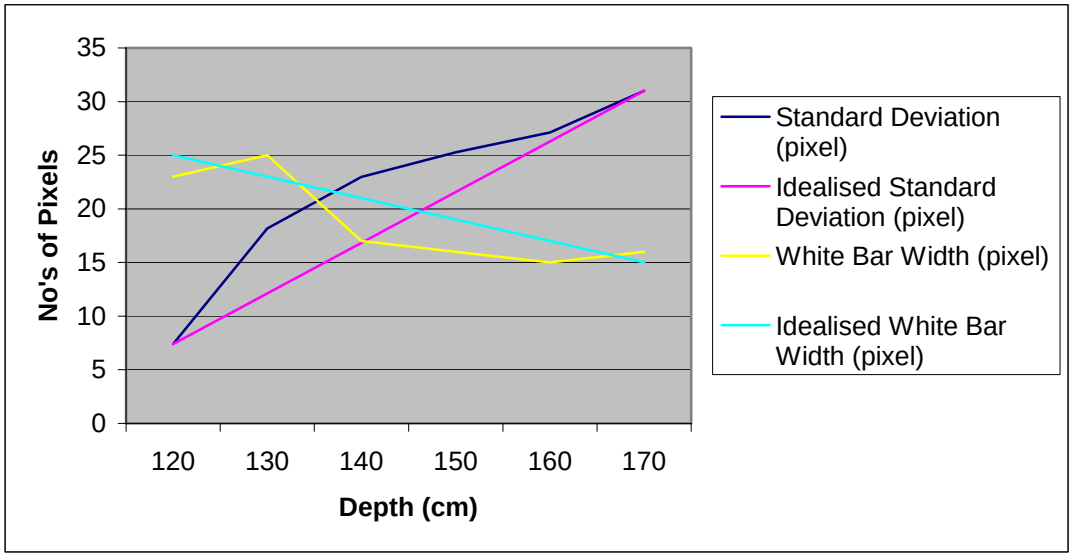


Figure 4.4 Plot of Depth Cues versus Depth

It can be seen from the results shown in Fig.4.4 that – for this limited experiment – the variation in width of the white bars due to depth-from-defocus was relatively close to the idealised requirement (i.e. relating a defocus cue to actual depth), but in fact it was not very linear. The standard deviation of the profile of the central white

bar, also did not offer a direct relationship to the idealised function required as a reliable depth cue.

A number of improvements in the set-up of the camera and projector arrangement (Fig 4.1) were made in order to produce better quality depth images. However, no improvement in the relationship between the first two depth cues and actual depth was recorded. A later set of improved depth images – this time varying in depth over steps of 5 cm – was used to investigate the remaining two possible depth cues (c & d from the list on p4-5). A further modification of the 'VertStripeCalc.m' program allowed these two depth cues (pattern intensity and mark/space ratio intensity) to be investigated.

Figure 4.5 shows the results of plotting the variation in pattern intensity due to depth.

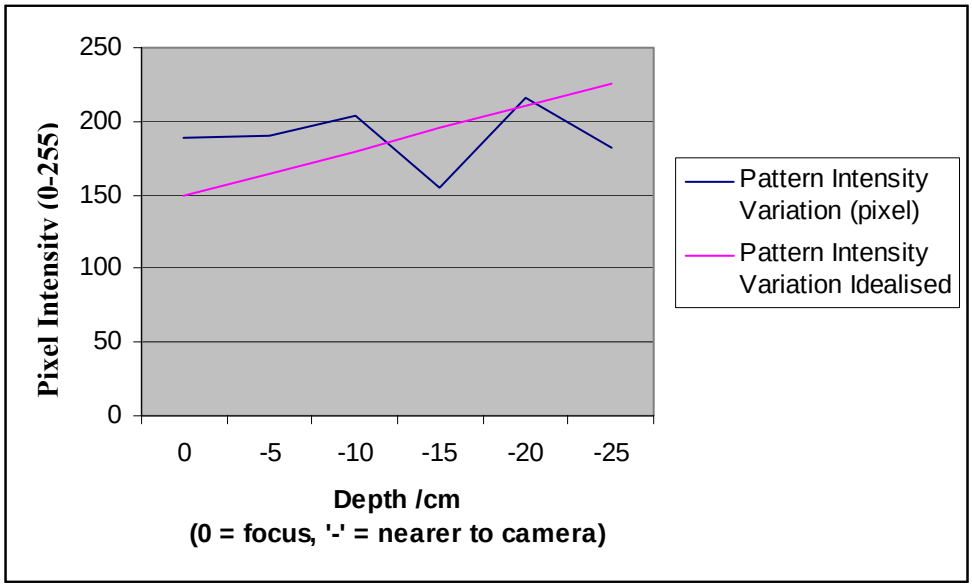


Figure 4.5 Plot of Pattern Intensity versus Depth

The hypothesis behind assuming that light intensity variation may relate to depth is simply that the intensity level would increase as the object plane is moved nearer to the projector and camera. However, as Fig 4.5 illustrates, the intensity variation did not relate well to the idealised form of response required. The most likely cause of this is the effect of defocus as the depth reduces. The intensity level of the peaks

representing the white bars of light in Figure 4.3 would tend to increase as the object plane moves nearer to the camera/projector and more of the light energy scattered from the surface is captured by the lens. However, as the light also becomes more defocused under these conditions, the light energy through the lens is dispersed (i.e. the edges of the peaks in Figure 4.3 become less vertical), and hence the peak intensity level is reduced. These two effects tend to counteract with respect to the relationship between depth and peak intensity level, making the latter an ineffective feature to use as a depth cue.

Fig 4.6 shows the resultant plot of the variation of mark/space ratio (ratio of white bar width to black bar width) against depth.

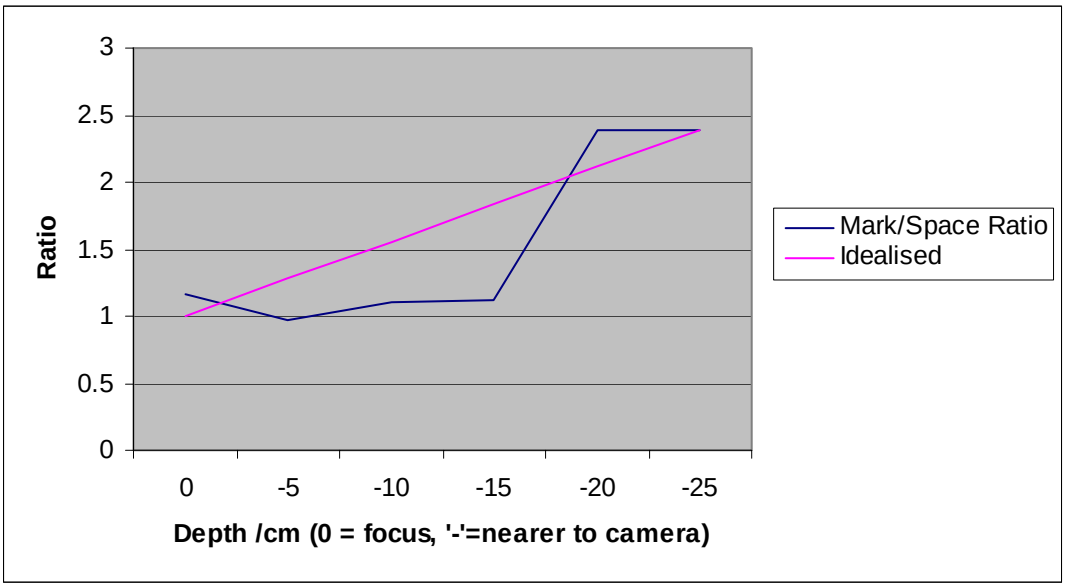


Figure 4.6 Plot of Mark/Space Ratio versus Depth

Whilst the mark/space ratio of white bar width to black bar width does increase with defocus – and reduced depth – as expected, it also does not vary in a consistent, monotonic manner that would make it a reliable depth cue.

4.4 Experiment 2: Investigation of depth cues in projected structured light patterns using a Foculus firewire camera

Having acquired a Foculus Firewire CCD array camera, it was possible to obtain more consistent, better quality images of the structured light pattern being projected onto a planar screen at variable depths. The recorded values and plots for this experiment can be viewed in Appendix B-2 (pp B-3), with the tabulated results on page B-9.

This experiment considered four possible depth cues. Specifically

a. Light intensity level (Grey scale levels)

b. Standard deviation of the extracted profile intensity levels (grey scale)

c. The period of the pattern frequency (pixels)

d. The mark/space ratio (white bar width to black bar width ratio)

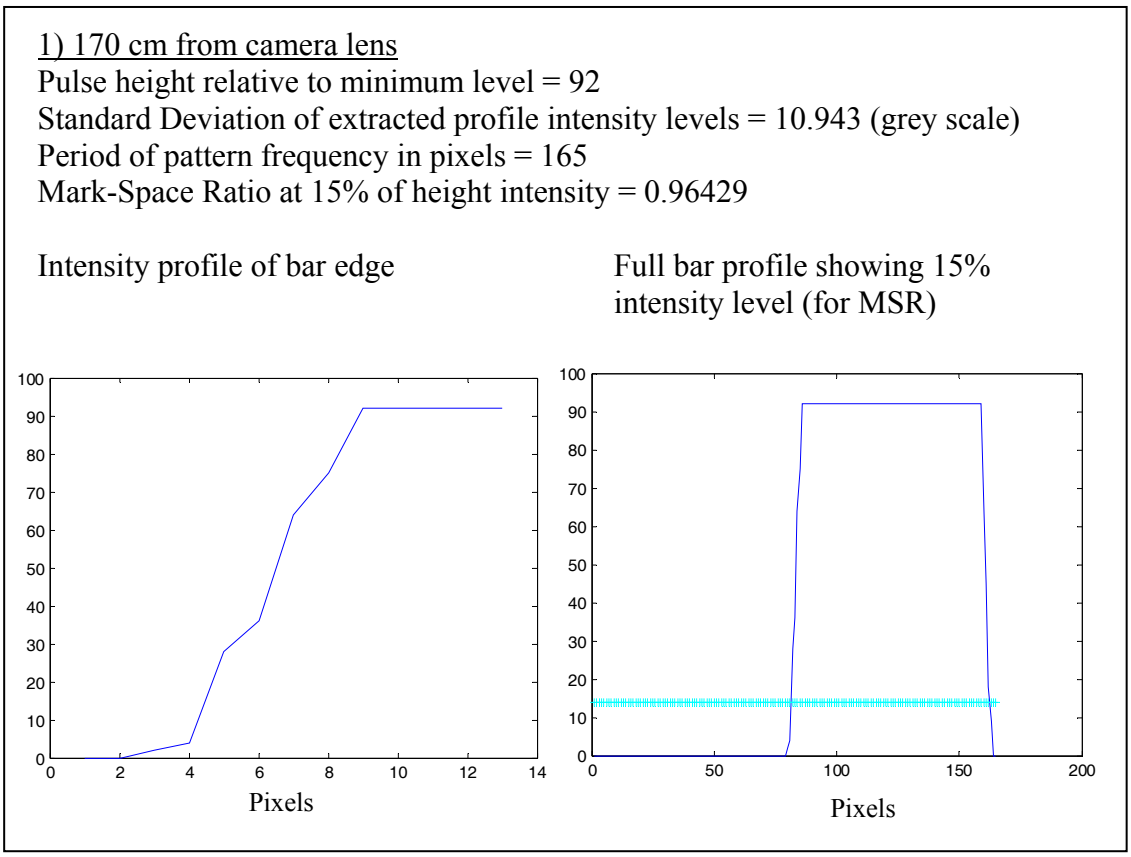


Figure 4.7 An Extract of Results from App B-2

For this experiment, twelve images were acquired over the range 170 cm to 115 cm at intervals of 5 cm. The point of focus (camera and projector) was at approximately 145 cm from the camera lens. Hence, the defocus effects were being monitored on either side of the point of focus (see tabulated results on page B-9).

For the four measured variables, depth cue (a) – the light intensity level – was immediately discounted as it remained at 92 Grey level points between maximum and minimum at all depths.

For (b) - standard deviation of extracted profile intensity levels – this only varied by approximately 1.5 greyscale values over the depth range of 55 cm (Fig 4.8), and the form of the response due to depth variation was non-linear.

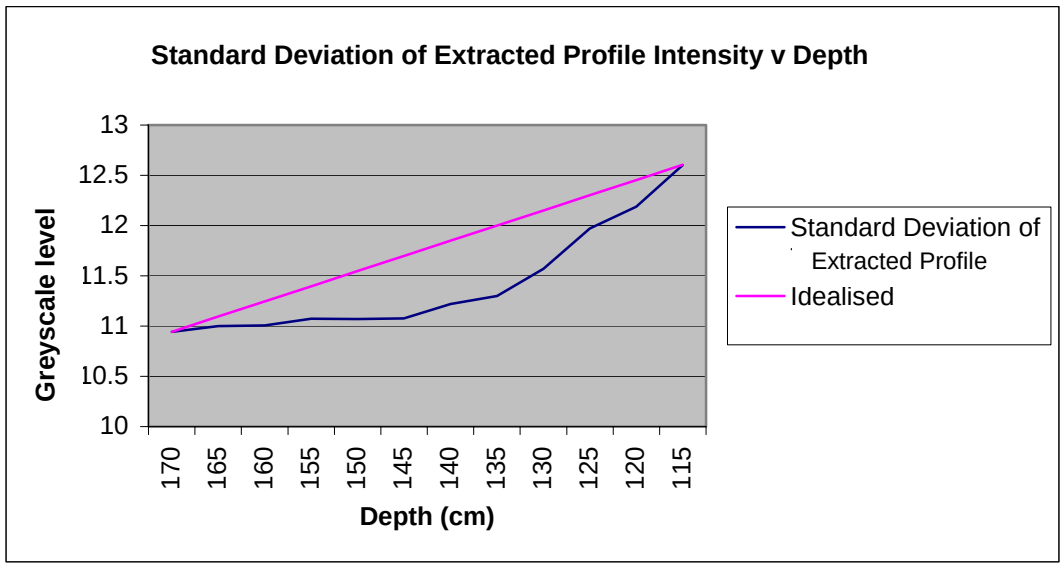


Figure 4.8 Plot of Standard Deviation versus Depth (Foculus)

For (c) - the period of the pattern frequency – the response is shown in Fig 4.9. The form of this response can be seen to be very linear over the depth range tested. However, the potential depth resolution offered by this variable is relatively limited. For a depth range of 55 cm the pattern period varies by 35 pixels (165 to 130 pixels), and hence a variation of one pixel in the pattern period would only represent a variation of approximately 1.57 cm in depth.

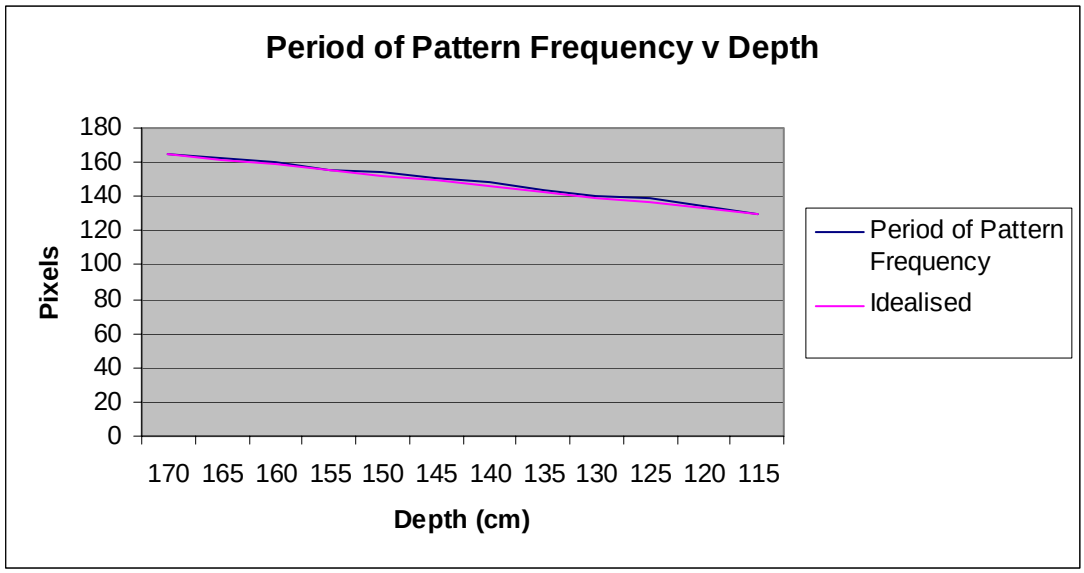


Figure 4.9 Plot of Pattern Frequency Period versus Depth (Foculus)

For (d) – mark/space ratio (white bar width to black bar width ratio) – a plot of the variation of this parameter over the depth range is shown in Figure 4.10. It can be seen that the value of mark/space ratio is 1.0 at a depth of approximately 145 cm. This is an expected result, as the pattern of vertical black/white bars was projected with an even mark/space ratio, and at a depth of 145 cm the pattern was in focus with respect to both the projector and camera. It was expected that the light energy forming the white bar would 'spread' due to defocus as the depth varied in either direction from the point of focus. However, it can be seen that the mark/space ratio did not provide a useful linear response to change in depth variation.

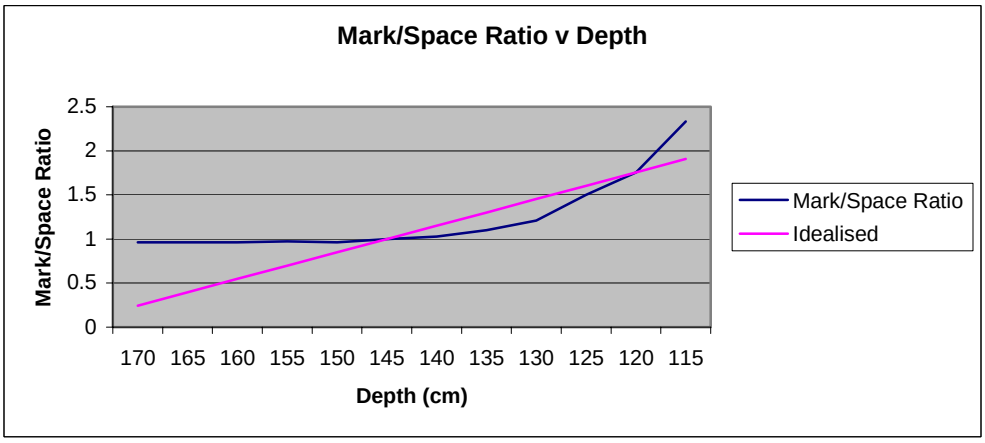


Figure 4.10 Plot of Mark/Space Ratio versus Depth (Foculus)

4.5 Intermediate conclusion on depth cue performance

From analysis of the results that have been discussed in the previous sections of this chapter (and recorded in Appendices B-1 and B-2), it is apparent that – of the depth cue parameters analysed – only the period of pattern frequency (Fig 4.9) appears to show the type of response that would make it useful as a variable parameter that can effectively relate a defocus effect to a measurement of depth.

This particular depth cue is related to the methodology used by Girod and Scherock [1], but can be seen to show similar limitations. Although the response is reasonably linear, the level of depth resolution that can be achieved is relatively low. A typical change in the measured period (ΔP) was approximately 4 pixels per 5 cm of depth change. Hence, a variation of $\Delta P = 1$ pixel, would only resolve a depth change of approximately 5/4 = 125 mm. Added to the low horizontal spatial resolution (number of depth estimates per row) that is limited by the number of pattern periods present in the image, makes this particular parameter unattractive for further development as a potential depth cue in a DfD system.

4.6 Experiment 3: The analysis of edge profile variation versus depth change

As a result of noting the variation in the shape of the edge profile (black bar to white bar transition) of the projected light pattern during depth change in Experiment 2, it was considered that this would be worthy of investigation as a possible depth cue that may be capable of better performance than those analysed in the previous two experiments. Appendix B-2 shows full results for the depth range 170 to 115 cm, but Figure 4.11 shows the edge profile variation at 10 cm intervals for the purpose of illustration.

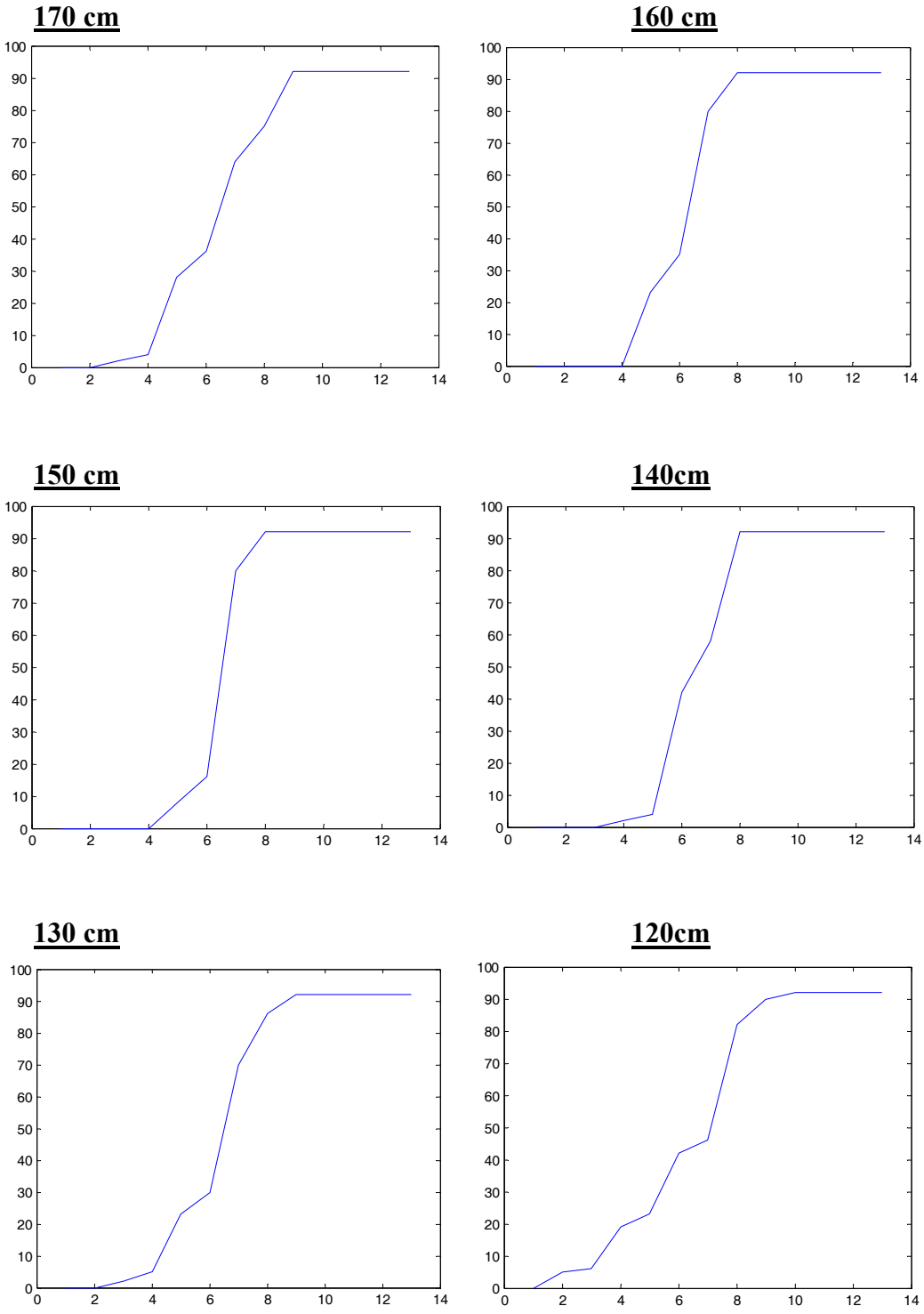


Figure 4.11 Variation of Edge Profile versus Depth

The point of focus (camera and projected pattern) was approximately 145 cm. It can clearly be seen that the profile is relatively vertical near this depth range. As the depth varies in either direction away from the point of focus, the incline of the edge

profile increases and more image pixels are required to describe the shape of the profile. This effect was considered worthy of further investigation, on the basis that it may provide a means of achieving greater resolution and accuracy of depth estimation over a given depth range.

4.6.1 Equipment for Experiment 3

Improved equipment was obtained for the purpose of carrying out this experiment. A high-resolution camera (UEye 1600 x 1200; 50mm lens) and data projector were fitted to an optical bench in a manner similar to that shown in Figure 4.1, except the camera was mounted directly to the optical bench (for stability and accuracy of measuring the camera-to-object distance) with the projector mounted above it. A vertical white planar screen was also mounted to the optical bench via a sliding track arrangement that allowed accurate horizontal positioning of the object screen relative to both camera and projector. The lens of both the camera and projector were aligned exactly vertically to each other in order to ensure that light distribution was projected evenly with respect to the screen and the camera's image of the screen. The projector had built-in adjustment for any keystone effects that would have distorted the projected image.

4.6.2 Modified algorithm for detection and extraction of edge profile

An algorithm was developed to detect each leading edge (black bar to white bar transition) as each pixel was examined in left-to-right order along each row of the image[4.1]. The algorithm determines the average greyscale intensity value for the entire row and uses this as a measure of whether the greyscale value of each pixel is within the position of a 'black' bar or a 'white' bar of the image of the projected structured light pattern.

[4.1] This algorithm was later used in a number of program functions, but can be seen in Appendix A-3

The methodology of this algorithm is illustrated in Figure 4.12. It can be seen that this is a variation of the algorithm developed earlier (see Figure 4.3 pp 4-4).

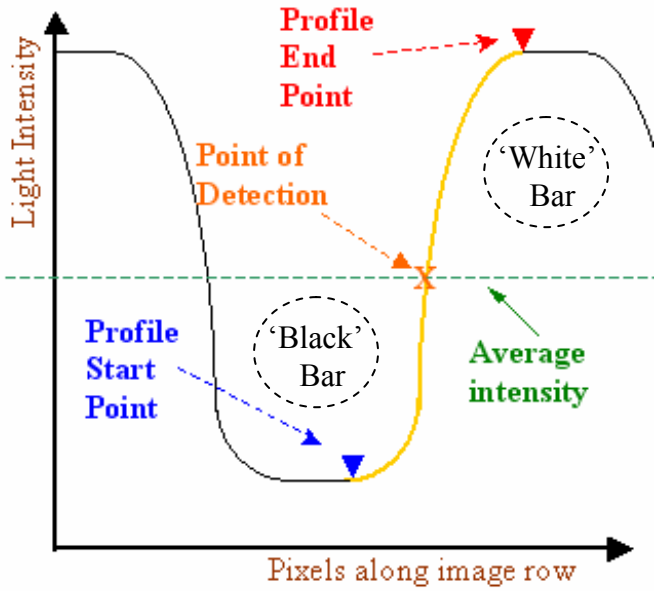


Figure 4.12 Detection and Extraction of Edge Profile

In order to avoid detecting profiles that may be incomplete (i.e. at start or end of the row), the algorithm first checks whether the pixel value at the start and end of the row is within a 'black' or a 'white' bar zone. If the first pixel on the row is a 'black' bar pixel, the following rising edge is ignored and the next rising edge is the first to be analysed. Similarly, if the last pixel on the row is a 'white' bar pixel, then the rising edge immediately preceding it is ignored, and the rising edge to the left of that is taken to be the last to be analysed[4.2].

For each valid rising-edge profile detected via the left-to-right search along the row, the point of detection is shown in Figure 4.12. From this point, the algorithm searches both to the left and right until it determines the local minimum and local maximum positions that will form the start and end points of the profile to be extracted.

[4.2] This reduces the number of potential depth estimations per row, but later programs develop a unique methodology to overcome this problem.

An important aspect of the profile extraction algorithm was the application of a smoothing filter in order to remove 'glitches' in the profile under investigation.

Figure 4.13 shows an example of the type of 'glitch' that could cause the algorithm to malfunction when determing the limits of the profile to be extracted.

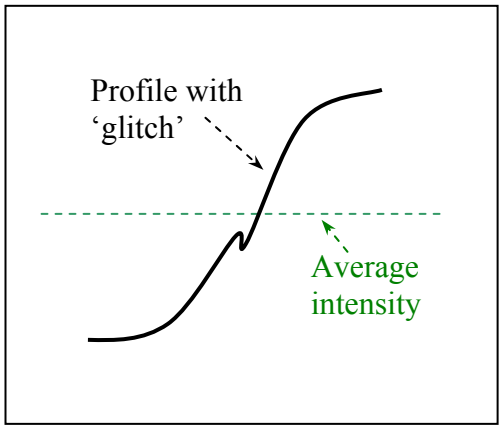


Figure 4.13 'Glitch' in Edge Profile

A 'moving average' type of smoothing filter was applied to the entire row profile prior to extraction of the edge profiles.

For example, if the response data for the complete row is labelled 'profile1', then the smoothed response data (e.g. 'profile2') uses five pixels as the default 'span' of data points to be averaged. For, e.g., the first four elements of the row response vector 'profile1', the algorithm averages as shown below.

The first four elements of profile2 are given by profile2(1) = profile1(1)

profile2(2) = (profile(1)+ profile1(2)+ profile1(3))/3

profile2(3) = (profile1(1)+ profile1(2)+ profile1(3)+ profile1(4)+ profile1(5))/5

profile2(4) = (profile1(2)+ profile1(3)+ profile1(4)+ profile1(5)+ profile1(6))/5

As this 'moving average' type of smoothing filter was only acting on a spread of five neighbouring pixels per averaging calculation, this did not have the effect of desensitising the depth measurement. The characteristic profile of the edge is inherently preserved, and remains dependent upon the degree of defocus. In any case, provided the same smoothing filter is applied to both the 'standard' edge profiles that are stored in the look-up table and the subject edge profiles being analysed, then any minor modifications to profile shape should cancel out.

4.6.3 Determining an edge profile parameter for evaluating depth

It had been considered desirable to design an algorithm that could compare a feature of an extracted edge profile against that of an identical feature for a profile of known depth. This could lead to a method of estimating depth that would have minimum computational complexity and minimum processing speed (The 'look-up' table method adopted by Ens & Lawrence [8] for a two-image system was an inspiration for this). The methods of profile comparison examined were

1. Curve fitting – correlating an element of a best-fit polynomial to the known depth of each 'standard' curve.
2. Simple summation – calculating the total of pixel intensity values under either the entire edge profile or a sub-section of the profile.

Method 1: Curve fitting

The first method investigated was that of curve fitting. A set of images of the vertical black/white bar pattern projected onto the movable planar screen was taken between depths of 115 cm (point of focus) and 80 cm, at 5 cm intervals (see Appendix B-3). The improved quality of these (high resolution) images can immediately be observed (by comparison with those in Figure 4.2 above). The 'spread' of the white bar (and hence the spread of the bar edge profile) can easily be observed by eye as the defocus effects increase over the depth range of 115 cm to 80 cm.

As with Experiment 2, the Matlab program 'VertStripeCalc.m' (App A-2) is used to sample a pixel row from the image, and then to isolate one period of the pattern and the profile of the rising edge of the white bar. Experimental trials of fitting various degrees of polynomials to the extracted edge profiles determined that the use of a

cubic polynomial[4.3] appeared to provide an acceptable level of fit without the potential complication of working with higher-level polynomials. (i.e. The need to determine a suitable metric that could be related to the change in depth would become computationally more complex as the order of the polynomial increased). The full results of this exercise can be seen in Appendix B-3, but a summary of the derived cubic polynomial equations can be seen in Figure 4.14 below.

Depth (cm)	Best Fit (cubic polynomial)
115	$y = -5x^3 - 0.39x^2 + 54x + 1.2e+002$
110	$y = -9.7x^3 - 4.2x^2 + 69x + 1.4e+002$
105	$y = -11x^3 - 0.17x^2 + 74x + 1.4e+002$
100	$y = -7.1x^3 + 2.1x^2 + 62x + 1.4e+002$
95	$y = -4.6x^3 + 5.6x^2 + 56x + 1.3e+002$
90	$y = -5.1x^3 + 1.7x^2 + 53x + 1.5e+002$
85	$y = -2.1x^3 + 2.2x^2 + 46x + 1.5e+002$
80	$y = -1.7x^3 + 1.3x^2 + 41x + 1.6e+002$

Figure 4.14 Best fit cubic polynomials for defocused edge profiles

Analysis of curve fitting as a potential depth cue

By observation of Figure 4.14, the cubic polynomial equations do not offer any direct correlation between the actual depth and any aspect of the expressions themselves. The possibility of deriving such equations as 'depth standards' (i.e. derived from known depths) then making an indirect relationship between them and similar equations fitted to the profiles of edges of unknown depth was also considered. Figure 4.15 illustrates a simple principle whereby a pixel (***x***) representing a known point in a profile under test can be evaluated for the value of its greyscale intensity (***I***). This could then be compared to the respective value of the same value ***x*** as derived from one of the cubic polynomial equations. The comparison would be repeated as an iterative function against each 'depth standard' equation until a matching – or best approximation – to ***I*** is found. It could also be repeated for more than one pixel position in the profile.

[4.3] MATLAB function 'polyfit.m' uses a least-squares method to generate the required coefficients

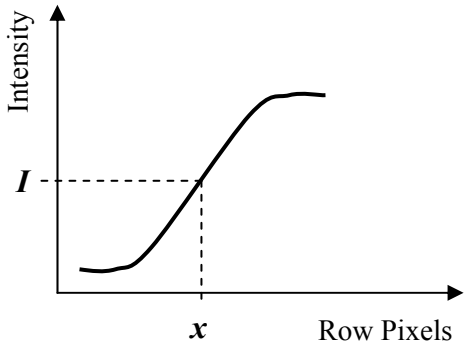


Figure 4.15 Illustrating an Indirect Method of Profile to Polynomial Comparison

In practice, such a method would be very difficult to implement. In order to ensure that a chosen pixel position (or vector element) from the test profile did accurately correlate to the same pixel position in the original 'depth standard' profile, very stringent constraints on the method for generating the edge profile would be required. Normalisation in terms of '***I***' would not be a problem, but in terms of '***x***' it would be far more difficult. The extraction of the edge profile would naturally cover a varying number of pixels (and hence vector elements) as the projected feature defocused and the amount of edge spread increased. Normalising in ***x*** would be problematic in terms of ensuring that the resultant profile still represented the same depth information.

It was determined that a possible method of relating curve fitting to depth in a fast, robust algorithm that was not computationally expensive would not be easily derived. By further empirical investigation of typical depth related profiles a far more accessible method was conceived (see Method 2 below).

Method 2: A simple summation method

Figure 4.16 shows a set of edge profiles and illustrates the profile variation over a depth range of 17 cm at 1 cm intervals.

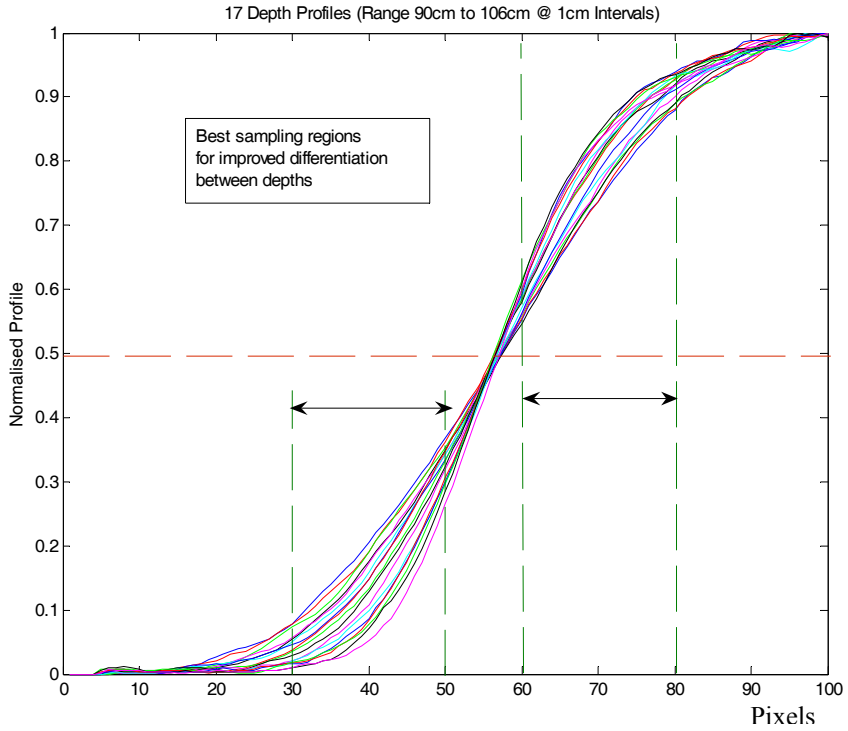


Figure 4.16 Analysis of Optimum Depth Cue Features

The profile plot that represents the edge feature in its most focused position is coloured pink in Figure 4.16, and can be seen to be furthest right in the lower half of the figure. Being in focus, this plot would – in theory – have the minimum gradient in its linear (central) region, and would appear as the leftmost plot in the curved section in the top half of the figure. Further plots – representing the increase in defocus as the depth range decreases – would present an increase in gradient and a corresponding variation in the position of the non-linear region of each plot (through to the dark blue plot that is leftmost on the bottom half of the figure, and rightmost on the top half).

Although not clearly visible in Figure 4.16, it was noted that the variation in edge profile at 1 cm intervals of depth range did not always follow such a precise pattern. The curved sections of the plots in the top half of the figure particularly illustrate this

fact. A general observation of these profiles led to the hypothesis that a simple summation of all – or part – of the profile values would provide an adequate means of identifying a particular profile and its relation to depth.

In order to test this theory, a new series of edge profile vectors were recorded for depths in the range 90 cm to 114 cm inclusive (i.e. 25 vectors at intervals of 1 cm).

The 25 vectors were stored in a look-up table, and a subsequent test image of the projected pattern at 102 cm was recorded, with an edge profile extracted from this image for the purpose of comparison with the look-up table.

The first method used the L1 norm to determine the residual error ($\boldsymbol{\varepsilon}$) between each element ($\boldsymbol{i}$) of the test vector ($\boldsymbol{t}$) and the stored vector ($\boldsymbol{s}$). Hence,

$$\boldsymbol{\varepsilon}_1 = \sum_{\boldsymbol{i}=1}^{N} \left| \boldsymbol{t}_i - \boldsymbol{s}_i \right| \qquad \{4\text{-}1\}$$

where $\boldsymbol{N}$ is the number of elements in each profile vector.

The results of this experiment can be seen in Figure 4.17 below.

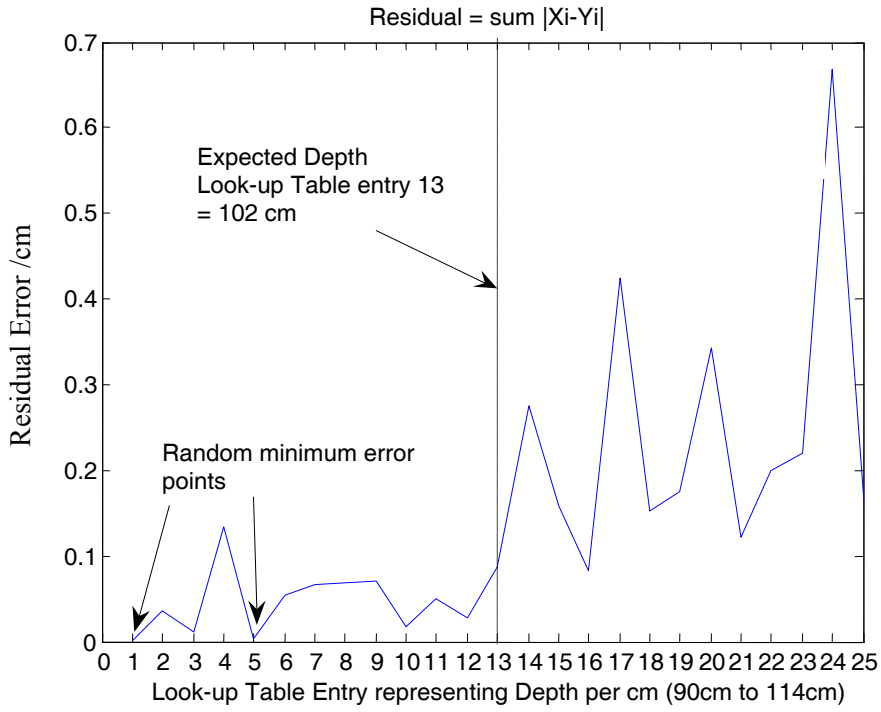


Figure 4.17 Comparison of L1 norm residuals

The results presented in the plot of Figure 4.17 did not show any correlation between the profile under test (representing a depth of 102 cm) and element 13 of the look-up table that holds the 'standard' vector for a depth of 102 cm. A residual error at – or close to – zero would have been expected at this point in the plot, with an increase in residual error as the test vector was compared to other vectors progressively further away from element 13.

The second method used the L2 norm to investigate residual error in a similar manner. Hence,

$$\varepsilon_2 = \sum_{i=1}^{N} (t_i - s_i)^2 \qquad \{4.2\}$$

The results for this experiment can be seen in the plot illustrated by Figure 4.18.

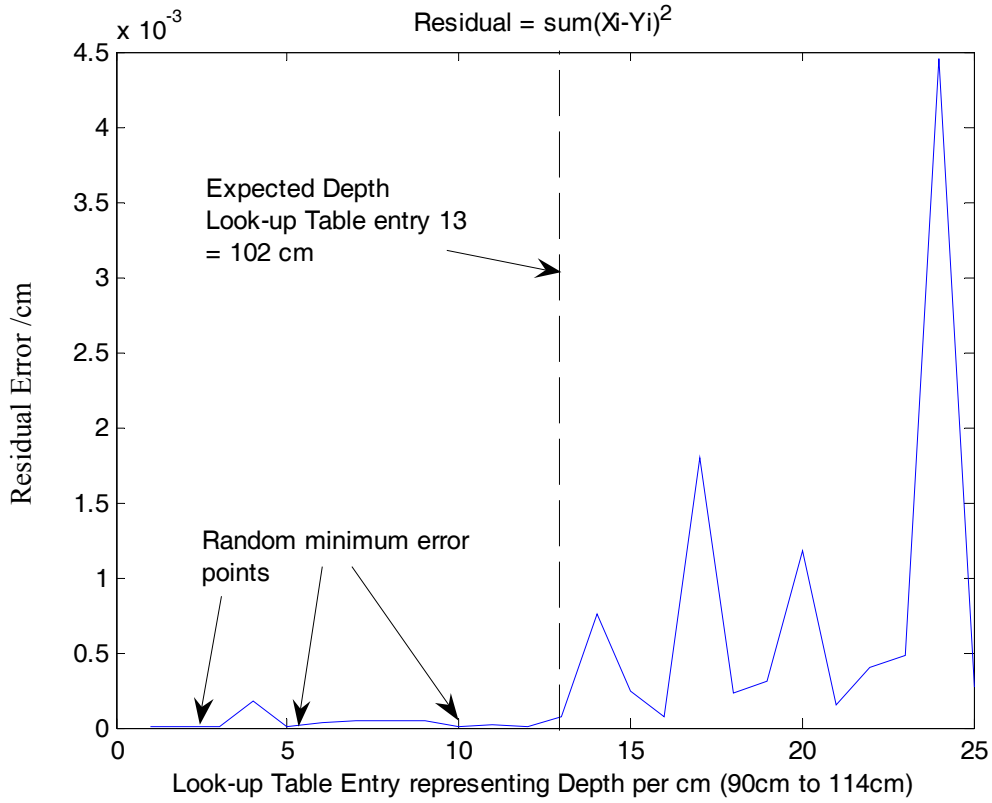


Figure 4.18 Comparison of L2 norm residuals

The results of this experiment also failed to produce an obvious means of correlation to the correct vector in the look-up table. However, the third method to be investigated simply used the difference between the sums of the two vectors under comparison. Hence,

$$\varepsilon = \sum_{i=1}^{N}(t_i) - \sum_{i=1}^{N} s_i \qquad \{4.3\}$$

The plot showing the results of comparing the test vector against each vector in the look-up table is shown in Figure 4.19, and it can be seen that it generated a remarkably good correlation to the set of look-up table vectors, and correctly generated a near zero value for the required depth vector.

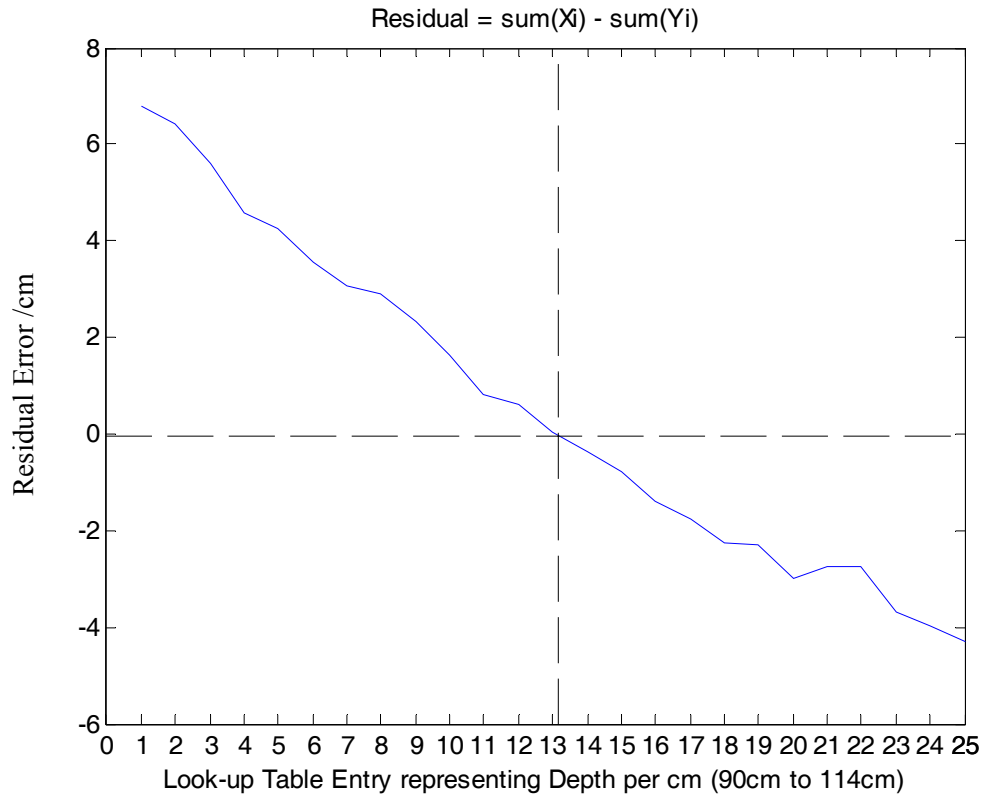


Figure 4.19 Comparison of Sum-Difference residuals

This method was subsequently adopted for all future development of a DfD system using projected structured light as described in Chapter 5. However, during these later stages of development it was found that the method could be made more robust by restricting the values used in the summation of each vector to those occurring between the vector elements of 30 to 50. Reference to Figure 4.16 shows that this section of the plot more than any other appears to display greater distinction between the depth vector plots. Hence, the algorithm can more reliably differentiate between stored vectors representing different depths.

4.7 Summary of Chapter 4

This chapter has described how features of a projected structured light pattern have been examined in order to observe how they vary with the depth of the surface on which the pattern is imaged. Features that have the property of variation that relates to depth are referred to as 'depth cues'.

The depth cue that provided the most useful response to variation in depth was the change in profile of the black/white edge transitions on an alternate black/white vertical bar pattern (Figure 4.2). It is proposed that a 'standard edge profile' that represents each depth can be stored in a look-up table, and edge profiles from objects under analysis can then be compared to the 'standard' depths in order to determine a depth estimate.

A suitable method of comparing profiles (in order to determine the best match) was also examined, and a simple summation method was adopted. It was necessary to devise specific programming algorithms (App A3 – A7) that were capable of analysing an image of the projected light pattern, and extracting the relevant edge profile data. These algorithms form the basis of the analytical programs used in chapter 5 of this thesis.

Chapter 4 - References

[45] Girod B, Scherock S, "Depth from Defocus of Structured Light", *SPIE,v1194, Optics, Illumination, and Image Sensing for Machine Vision; pp 209-215*; 1989

[46] Hinojosa C, Serrano-Heredia A, Ibarra J, "Recovery of three-dimensional shapes by using defocused structured light", *Optics & Laser Technology; v30; p281-290*, 1998

[47] Pentland A P, Scherock S, Darrell T, Girod B, "Simple range cameras based on focal error", *J.Opt.Soc.Am; v11 No.11; pp.2925-2934*, 1994

[48] Nayar SK, Watanabe M, Noguchi M, "Real-time Focus Range Sensor", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *v18 No.12, pp1186-1197*, 1996

[49] Kowarschik_R, Kuhmstedt_P, et al, "Adaptive optical three-dimensional measurement with structured light", *OPTICAL ENGINEERING, v39, No 1, pp.150-58,* 2000

[50] Ghita O Whelan P F, "A Video-rate Range Sensor Based on Depth From Defocus", *Opt Laser Technol; v33, No3, pp167-176*, 2001

[51] Zhang L, Nayar S, "Projection Defocus Analysis for Scene Capture and Image Display", *ACM Transactions on Graphics 25 (3), pp. 907-915*, 2006

[52] Ens J and Lawrence P, "Investigation of methods for determining depth from focus ", *IEEE Transactions on Pattern Analysis and Machine Intelligence, v15 No2, pp. 97-108,* 1991

Chapter 5

The Development of an Active DfD System

Using, as a depth cue, the black-to-white edge profile of the vertical bar projected light pattern (as described in chapter 4), an active depth-from-defocus system was developed via a number of iterations and modifications. The development process is described in this chapter via the following topics

- The initial camera/projector arrangement.
- Development and evaluation of the pattern analysis algorithm and the associated 2-D look-up table method.
- Experiments to evaluate the effect on the performance of the 2-D look-up table method relating to
    - Variation in the light absorption properties of the object surface.
    - Variation in the angle of the object surface.
    - The use of time-lapsed image averaging.
- Potential problems caused by the use of a data projector.
- Improving horizontal spatial invariance of the depth estimates by moving the projected pattern during object analysis, and the associated development/evaluation of a 3-D look-up table method.
- Improving vertical spatial invariance via the development/evaluation of a 4-D look-up table method.
- Neutralising original image texture in order to improve depth estimate accuracy.

5.1 Initial set up and the fundamental pattern analysis algorithm

The initial camera/projector set-up is shown in Figure 5.1.

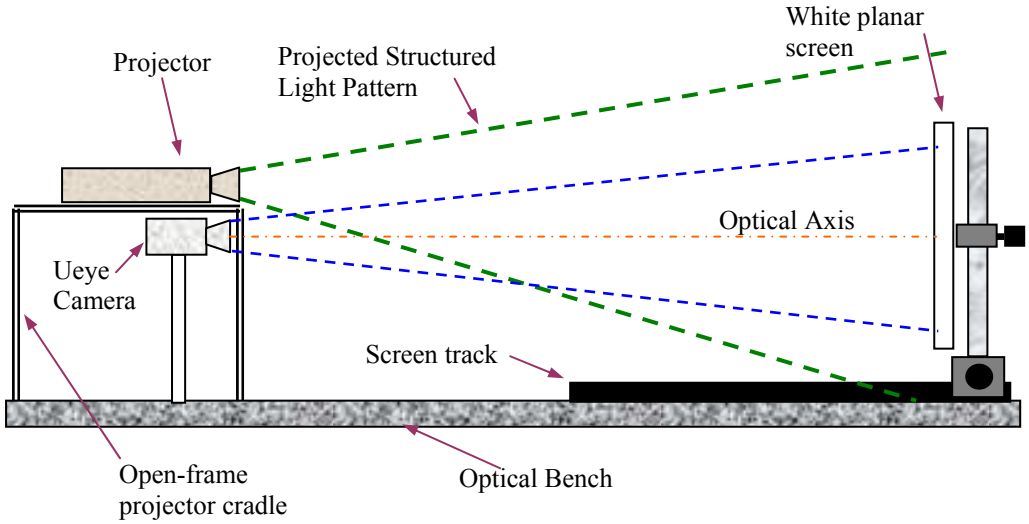


Figure 5.1 Initial Camera/Projector Set-up

All initial investigations using this set-up involved only the use of planar images of the projected light pattern for the purpose of both creating standard depth profiles in a look-up table and testing depth images against the profiles in the look-up table.

The data projector was connected to a laptop computer that was generating a vertical black/white bar pattern of the type shown in Figure 4.2. The camera and projector were centred horizontally, and the projector cradle could be adjusted vertically to ensure that the structured light pattern was set to the optimum vertical position in relation to the horizontal range of the planar screen. It can be seen from Figure 5.1 that the camera only viewed a section of the planar screen area. This meant that the entire field of view of the camera was filled with the image of the projected pattern irrespective of the horizontal distance of the planar screen relative to the camera.

An interesting feature (though not vital to the final design) of this arrangement was the fact that an almost perfectly telecentric relationship existed between the camera, projector, and position of the planar screen. With the screen at its furthest distance from the camera, the camera would view approximately 8.9 periods of the projected

pattern horizontally across its field of view. As the screen was moved nearer to the camera/projector, the pattern – being divergent from the projector – would appear 'smaller' on the screen. However, as it was physically nearer to the camera, it would naturally tend to appear 'larger' in the image. In practice these two effects cancelled each other almost perfectly, to the extent that approximately 8.9 pattern periods occurred horizontally in images captured at all depths, giving a constant horizontal pattern period of 180 pixels (horizontal image resolution being 1600 pixels).

In this initial set-up, the positions of the planar screen were measured manually in terms of distance from the camera lens. In later versions this task was automated, as will be discussed later in this chapter.

5.1.1 Development of the algorithm to generate a 2-dimensional look-up table of standardised depth images

The fundamental pattern analysis algorithm was based upon the illustration shown in Figure 5.2 below. The purpose of the algorithm was to analyse the image of the pattern at each depth, to extract the valid edge profiles from each row, and to average them to produce a 'standard' edge profile to represent that depth in a look-up table. The look-up table is a 2-dimensional array, one dimension being the vector holding the 100 elements of the normalised edge profile, the second dimension being the number of vectors corresponding to the number of depths imaged. In this case the number was 25 (90 cm to 114 cm inclusive at steps of 1 cm). So the 2-D look-up table generated was a 25 x 100 element array.

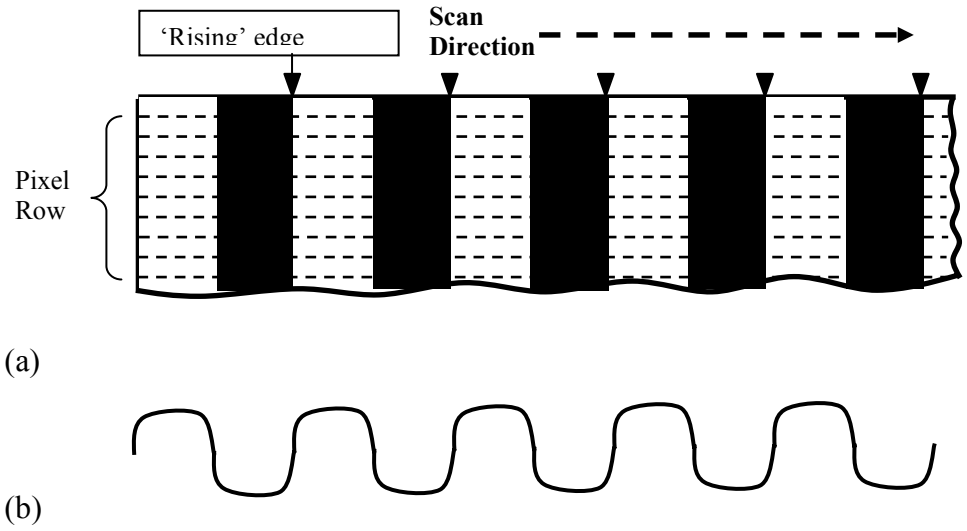


Figure 5.2 (a) Illustration of Captured Image Section

(b) Typical profile of intensity levels along a row of pixels

The algorithm (see 'AverageCalc.m', Appendix A-3) captures an image of the projected pattern from the surface of the planar screen at a known depth. Figure 5.2 (a) illustrates that each image row is scanned to extract a profile (as illustrated in Fig 5.2 (b)) of the 1600 pixel values for each row. The position of each valid profile of a black-to-white bar (or 'rising edge') transition on that row then needs to be determined.

Section 4.6.2 of chapter 4 (including Figure 4.12) dealt with the method of extracting an individual edge profile. This section deals with the methodology by which the entire image is processed and the 'standard' edge profile representing that particular depth is stored in the look-up table of depth standards. Figure 5.3 is a flowchart (covering two pages) that describes the programming algorithm.

Figure 5.3 illustrates how each row profile is analysed to locate and extract all valid edge profiles and average them into an accumulator vector. In a sense, this exercise represents a form of calibration of the system against images of known depth.

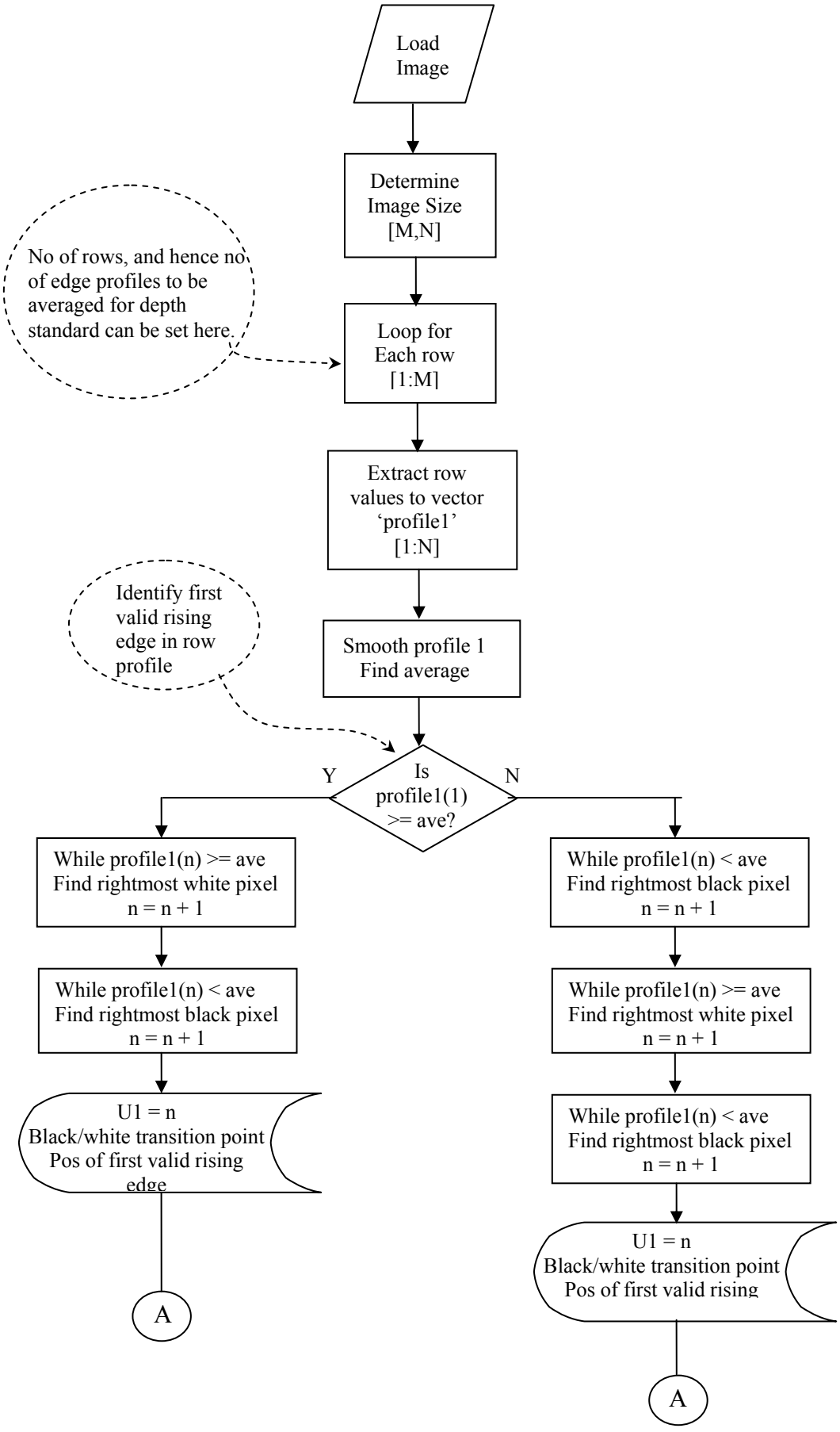


Figure 5.3 Flowchart to Illustrate the Algorithmic design of AverageCalc.m

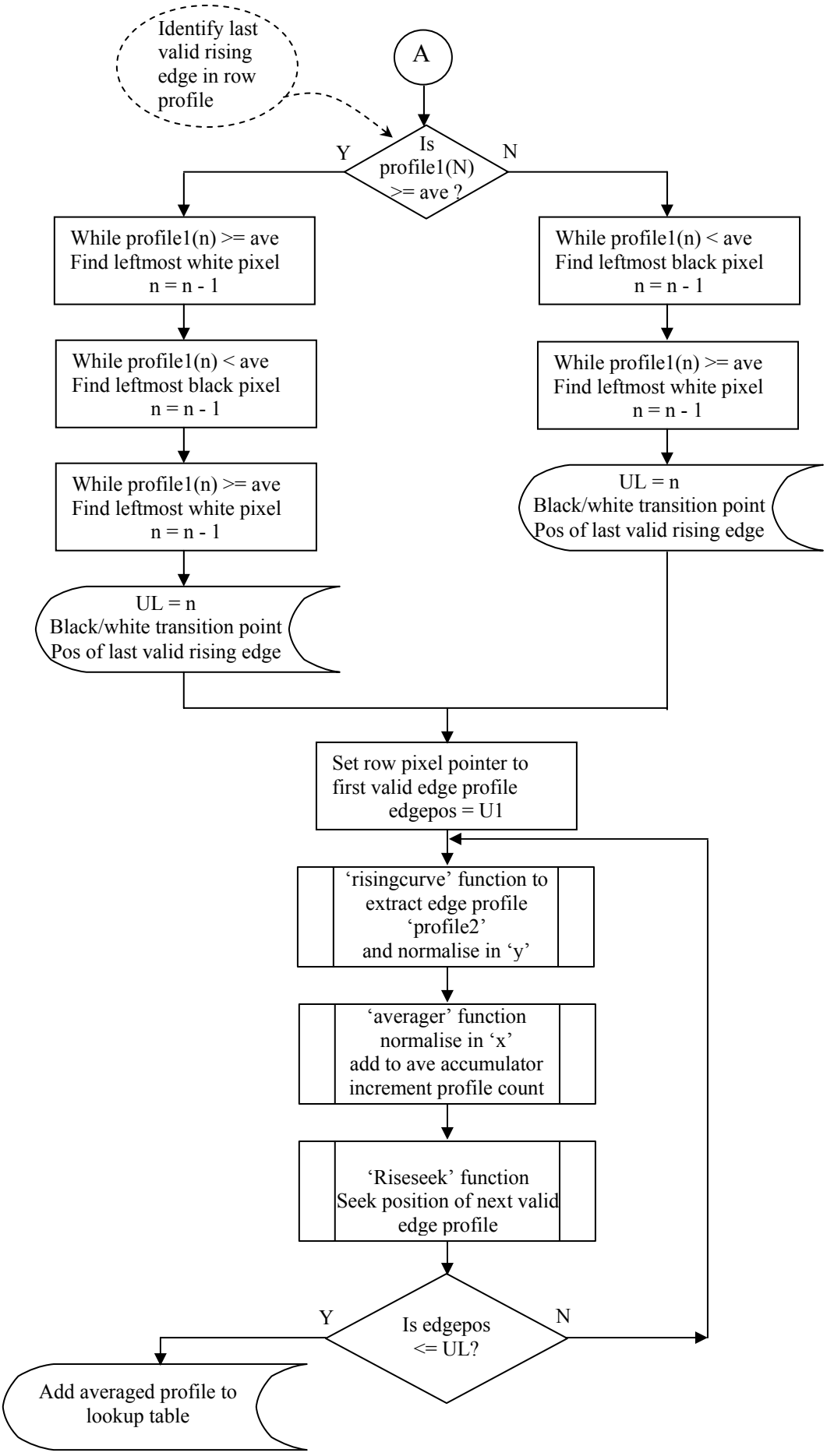


Figure 5.3 Flowchart to Illustrate the Algorithmic design of AverageCalc.m

The functions listed in Figure 5.3 can be viewed in Appendix A ('risingcurve.m' in Appendix A-4, 'averager.m' in Appendix A-5, and 'Riseseek.m' in Appendix A-6).

The sequence used to apply the above algorithm included,

- Set distance of planar object screen to e.g. 90 cm (minimum determined distance for camera/projector to produce suitable image at out of focus condition)[5.1].
- Project structured light pattern and capture image at that depth
- Analyse and store averaged profile as 'standard' for that depth (note 90 cm 'standard' stored at look-up table position 1, 91 cm at position 2, etc.
- Repeat for all depths at 1 cm intervals, up to and including 114 cm (point of focus for camera and projector).

It can be seen from the above sequence, that the first index of the 2-D look-up table (i.e. the position of the profile vector in the table) actually represents the depth measurement. In practice, the projector was left switched on throughout the entire sequence. In fact it was turned on several minutes prior to the start of the sequence in order to make sure that it was at its optimum operating temperature for all captured images (See "Data Projector Problems" later in this chapter).

It was now necessary to produce a program to test the 2-D look-up table method.

5.1.2 Development of an algorithm to match an edge profile with its nearest corresponding profile in the 2-D look-up table

Figure 5.4 illustrates the method by which an edge profile – representing an unknown depth – is compared with the standard profiles to find a 'best fit'. The position of the 'best fit' standard profile in the 2-D look-up table is a direct indicator of the depth range that the test profile represents. (See also Appendix A-7, ppA-12, for the actual profile-match program function).

[5.1] Figure 5.1 shows a natural limit to where the lines of projection and camera field of view intersect.

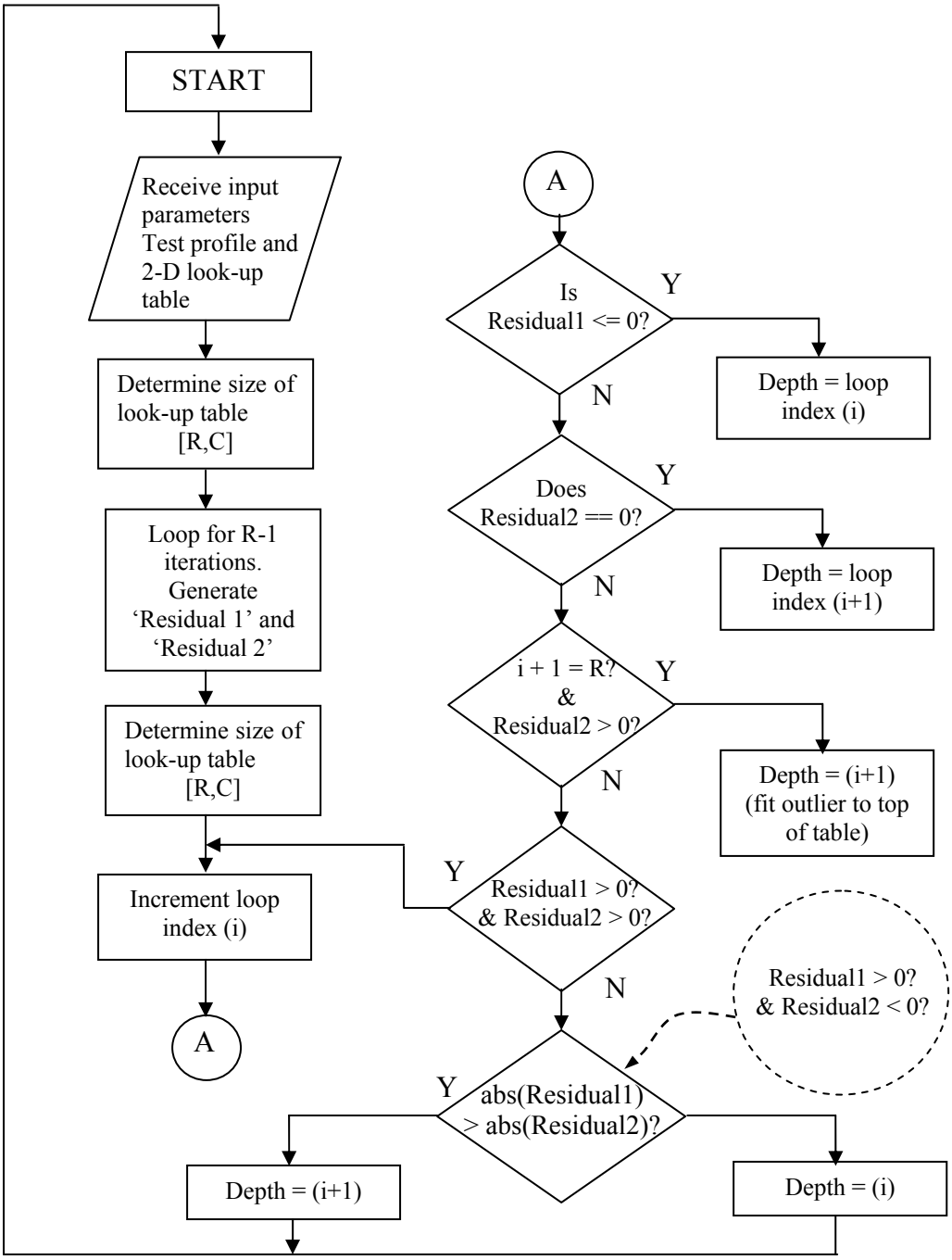


Figure 5.4 Flowchart of Depth Matching Algorithm

Figure 5.4 shows that the residual value generated by the edge profile comparison method (described in section 4.6.3) is used to determine how closely the test profile compares to each 'standard' profile in the look-up table. A feature of the algorithm is that it makes a 'double' comparison on each iterative loop. Because the look-up table values decrease monotonically as the edge becomes more in focus, the double comparison method uses relational operators to determine whether the test profile more closely matches the *current* look-up table position or the *next* look-up table position. If the initial operation shows that the test profile value is *less than* both of them, then no further 'best fit' operations are required, and the program can proceed to compare the test profile with the next look-up table pair.

Where the test profile value falls in between two profiles in the look-up table, then the program determines which standard profile is the best match and determines that table position to be the effective depth estimate.

5.2 Experiment 4: Evaluating the Depth-from-Defocus algorithm using a 2-Dimensional look-up table

The purpose of this experiment was to record an image of the projected pattern for each depth in the range 90 cm to 114 cm, as per the 2-D look-up table of standard depth profiles, and to determine the average depth estimate error for each of the 25 depths.

Each image was analysed using the 'profilematch.m' algorithm (Fig 5.4, App A-7) that was imbedded in the program 'ImageDepthCal.m' (App A-8). The results are displayed in Appendix B-4 and are summarised in Table 5.1 and Figures 5.5 and 5.6 below.

Depth	% Error	STD	Depth	% Error	STD
90	0.38154	0.34339	102	1.7242	1.7587
91	0.74048	0.67383	103	1.467	1.511
92	0.87974	0.80936	104	1.2619	1.3124
93	1.09	1.0137	105	1.366	1.4343
94	1.201	1.1289	106	1.4058	1.4901
95	1.1655	1.1072	107	1.5507	1.6592
96	1.0554	1.0132	108	1.3481	1.456
97	1.0583	1.0266	109	1.2804	1.3957
98	1.1974	1.1735	110	1.4029	1.5432
99	1.2807	1.2679	111	1.706	1.8937
100	1.4344	1.4344	112	1.4652	1.641
101	1.5876	1.6035	113	1.3338	1.5072
			114	1.6653	1.8985

Table 5.1 Results of First DfD Algorithm Test (2-D LUT)

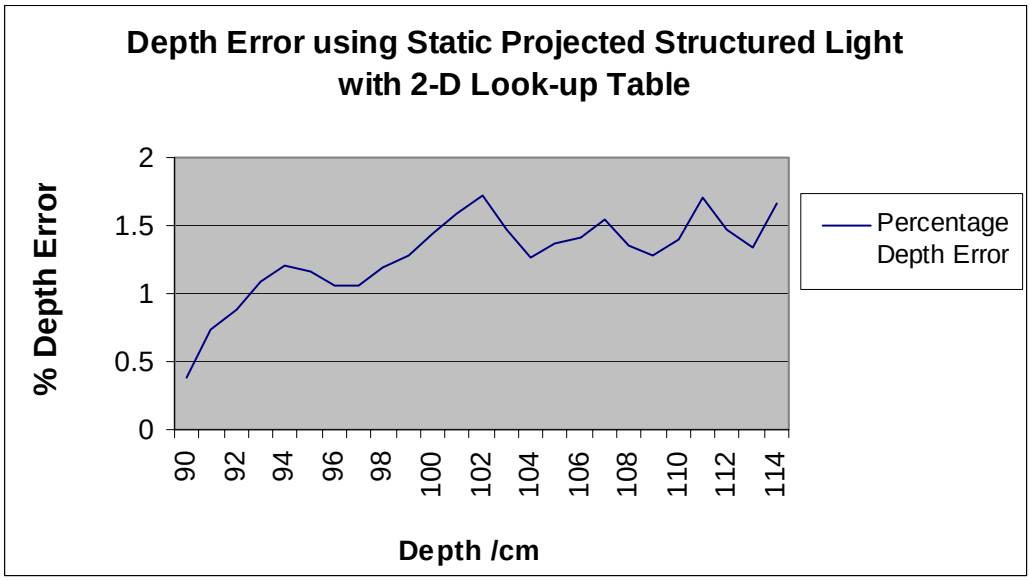


Figure 5.5 Plot of Error Results from Table 5.1

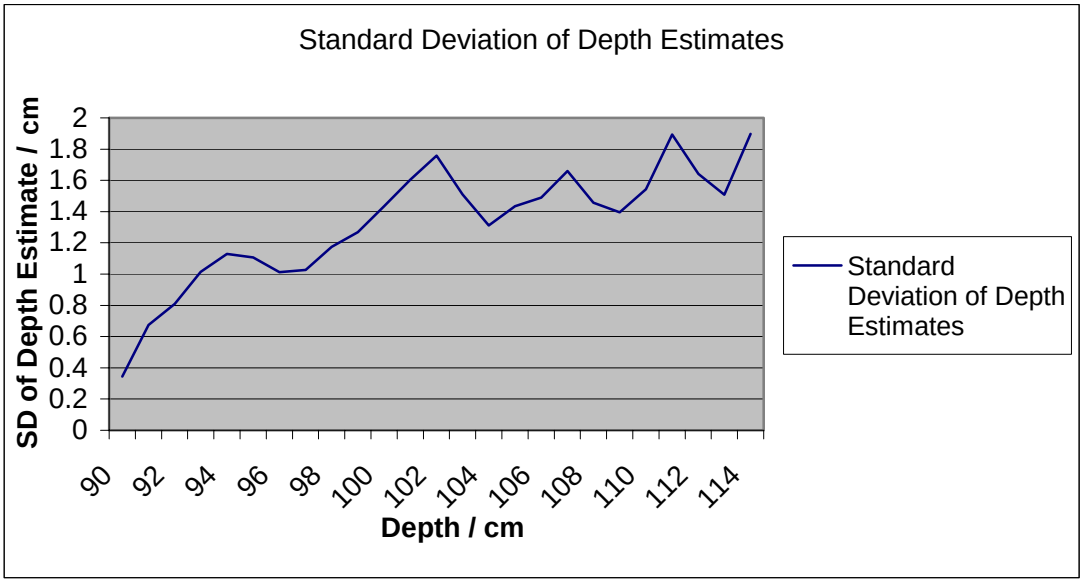


Figure 5.6 Plot of Standard Deviation of Depth Estimate Results from Table 5.1

The results shown in Figure 5.5 show that the depth error was less than 0.5% at the nearest depth of 90 cm. The error gradually increased with depth, but remained at approximately 1.5% even at the furthest depths in the range. Depth estimation accuracies of less than 2% are very good for a first attempt with a single-image DfD system (see Table 3.4, Ch3), and it was considered that further improvements could lead to the development of a useful tool for measuring soft objects and/or providing fast measurements. The standard deviation of the set of estimates per depth (Figure 5.6) also indicated that the spread of estimate values was relatively low and was approximately proportional to the accuracy of the error.

A limitation of this experiment is that a depth average over each entire planar object is used, and hence no evaluation of spatial variation over the image area is made. However, before beginning to assess – and develop improvements in – spatial variation, it was decided that further experiments would be carried out to determine how well the current 2-D look-up table system coped with variations in e.g. object surface angle, and surface reflectance. The following experiments document this work.

5.3 Experiment 5: Evaluating the effect of variance in surface reflectance

In order to determine how successfully the DfD algorithm would operate on surfaces of different reflectance, the structured light pattern was projected onto surfaces that had increasing levels of grey intensity, and hence increased light absorption. This meant that the height of the intensity step between the 'white' and 'black' bars of the pattern was gradually decreasing as the object grey level increased. As each profile was normalised in its intensity range prior to analysis, it was conceivable that the normalised profile would retain enough of its shape due to defocus to enable it to be accurately matched to a 'standard' profile from the DfD look-up table. All images were at a depth of 100 cm. The same program ('ImageDepthCal.m', App A-8) was used to analyse the images, and the full results are provided in Appendix B-5.

Six images were generated, each one being of the structured pattern projected onto backgrounds that were white, 25% grey, 40% grey, 50% grey, 80% grey, and black. These colours were taken from the standard colour range offered in the 'Microsoft Word' word-processing drawing tools. They may not have represented absolute accuracy of grey level percentage (particularly once printed), but as a relative value they were adequate for the purposes of this experiment. The results illustrated in Appendix B-5 show that the algorithm continues to operate reasonably successfully up to and including the image taken from a 50% grey background. However, it should be noted that – although the mean depth estimate values for images with backgrounds up to 50% grey level remained within approximately 2.5% of the actual depth – the standard deviation of all estimates per image increased monotonically as the background became darker.

A plot of the Standard Deviation and percentage error versus the variation in background grey level is shown in Figure 5.7 below. It should be noted that the plot – being of only six measurements – is rather discrete. However, it offers enough information to suggest that the algorithm will produce reasonably accurate results when presented with low contrast images.

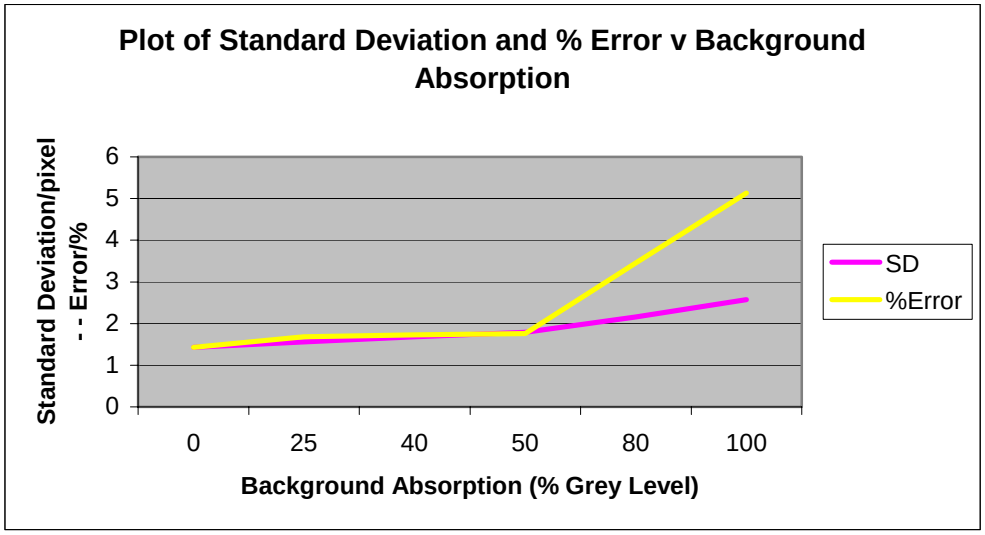


Figure 5.7 Plot of Standard Deviation and % Error versus Background Absorption

5.4 Experiment 6: Evaluating the effect of surface angle

As all experiments up to this point had used a planar object screen that was perpendicular to the optical axis, a measure of the performance of the algorithm when presented with an angled surface of reflection was considered desirable.

A white planar surface was set up at a distance of 96cm from the camera lens. The surface was positioned on the rotational axis of a shaft mounted in a vertical holder. The field of view of the camera was centred exactly on the rotational axis of the object, and the projected (vertical bar) pattern was positioned such that a rising edge was also directly located on the centre of rotation. (See Figures 5.8 and 5.9)

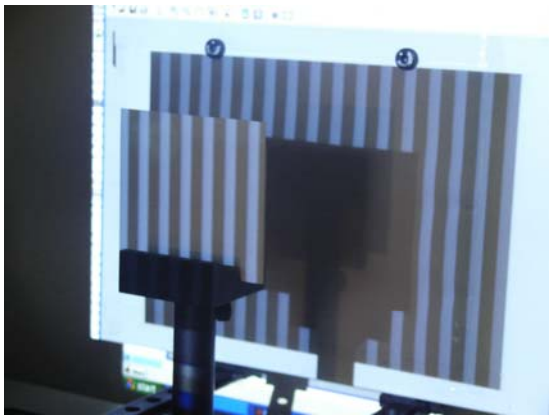


Figure 5.8 Surface Angle at 0^o

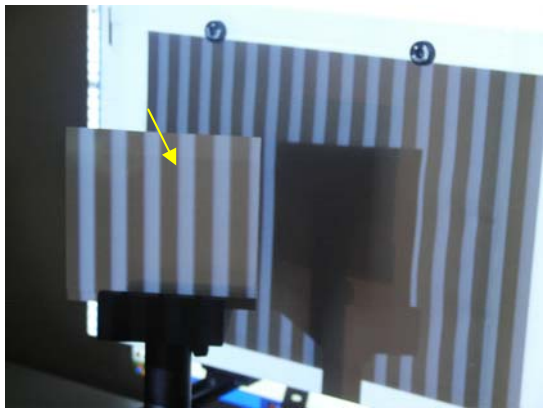


Figure 5.9 Surface Angle at 30^o

Figures 5.8 and 5.9 show the test surface angled at 0^o and 30^o respectively. The yellow arrow in Figure 5.9 illustrates the 'rising' edge that was positioned on the centre of rotation and used to estimate the depth at different angles. Clearly, the edge profile used for this was required to remain on the centre of rotation otherwise its actual depth would have varied as the rotation of the test surface varied.

An image was captured with the surface being first at 0^o to the optical axis, then 20^o, 30^o, 40^o, 50^o, and 60^o respectively. The results (Appendix B-6) and associated plots in Figure 5.10, show that the percentage error for surface slopes of up to 60^o of rotation from the camera axis are within accuracy bounds of 1.5 %. This also represents a satisfactory result in terms of the eventual level of performance that could be expected when the algorithm was applied to real world objects.

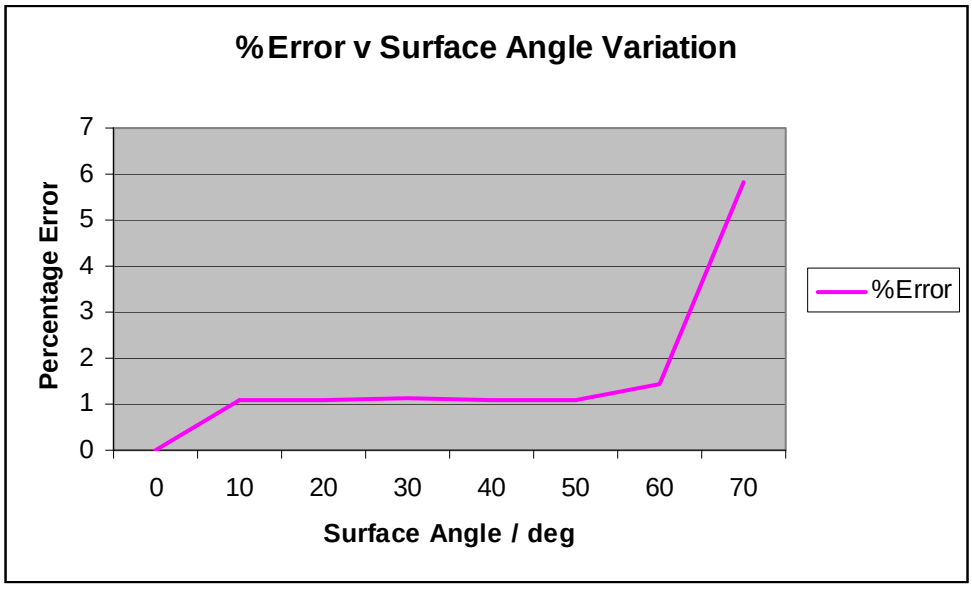


Figure 5.10 Plot of % Error in Depth Estimation versus Variation of Surface Angle

The plot illustrates a remarkably high tolerance to the degree of surface angle rotation up to approximately 60^{o}, though no readings were taken for angles greater than 70^{o} as it was clear that the image of the projected pattern was so distorted by this stage that the level of error would be extremely high.

5.5 Experiment 7: Evaluating the effect of image averaging

It is well understood in the field of image processing that averaging a number of time-lapsed images of the same scene can improve the signal-to-noise ratio. This experiment was carried out in order to determine whether image averaging would improve the accuracy of depth estimates over those recorded in Table 5.1 and Figure 5.5. Sixteen images of the structured light pattern were taken at time intervals of approximately 15 seconds (time to capture, label and save the image). The images were then averaged accumulatively, and were analysed via the DfD algorithm until all 16 images had been included in the averaged image. The results are shown in Appendix B-7 and are illustrated via the plot of Figure 5.11.

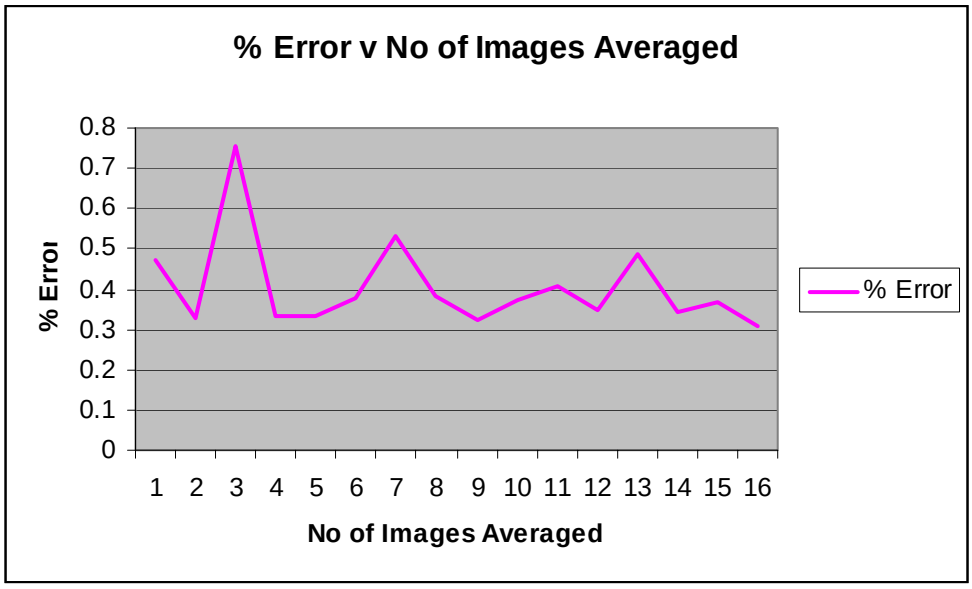


Figure 5.11 Plot of % Error versus No of Time-lapsed Averaged Images

The plot shows no particular trend of improvement in estimated depth error as a result of time-lapsed averaging, and does not suggest that there would be any benefit in including this process in the algorithm.

Reasons for image averaging not making any improvement could include

- The signal strength – being the projected pattern edge profile – is already so dominant compared to image noise that the averaging is largely ineffective, with other effects leading to measurement errors.
- The position of each edge profile is not perfectly static between time-lapsed images. Hence the profiles are not exactly aligned during the averaging process and tend to be corrupted rather than reinforced with respect to background noise.

One aspect that is relevant to the latter point is the use of the data projector (Plus U2 1110 – App C-1). This is a relatively high-specification data projector when used for its intended purpose, but when considering variations between time-lapsed images some of its potential limitations also became apparent. Although the data projector would offer considerable benefits in terms of how the DfD algorithm could be improved (see section 5.7 and general system development beyond that section), its

optical features could introduce unwanted effects on the projected pattern, and hence the image used for DfD analysis. As part of the development of the DfD system, section 5.6 of this chapter discusses features of the data projector that potentially affected the quality of results.

5.6 The Plus U2 Data Projector – Potential problem source

Throughout the development of the DfD system, the use of the Plus U2 data projector (App C-1) to project the structured light pattern became increasingly beneficial in terms of allowing movement of the pattern position during DfD operations (see later sections of this chapter). However, there were also occasions when inconsistencies in the performance of the system were difficult to explain, and it was considered likely that variability of the projected image produced by the projector was the most likely cause. This section of the chapter describes the potential problems caused by the data projector. The topic is dealt with at this stage of the thesis as the measures taken to overcome these problems actually led to significant developments in the DfD system methodology.

The exercise of generating the 2D look-up table by analysing sets of images at known depth in order to produce 'standard' edge profiles to represent each depth is a form of system calibration. Once generated, this exercise should be repeatable in the sense that an edge feature projected from the same structured pattern to the same depth should correlate to the profile representing that depth in the look-up table, and hence the depth is identified. In certain conditions it was found that this did not always operate as successfully as expected. As part of this chapter's explanation of the system development, it is worth noting some of the features of the Plus U2 data projector that were potential sources of error.

5.6.1 Data Projector Operating Temperature

The data projector uses a 150 W mercury vapour lamp that produces a dark blue glow when it is first turned on as only a small amount of the mercury is vaporised and the arc-tube gas pressure is relatively low. As the main arc strikes, and the gas heats up and increases in pressure, the light emitted is in the visible range and the high gas pressure causes the mercury emission bands to broaden. Once the lamp is at its optimum operating temperature it produces a light that appears more-white to the human eye. (The colour temperature can range from 4200K to 7500K dependent on the bulb design). Hence, ensuring that the lamp has fully reached its operating temperature is important to the quality of the image that is projected. This will also help to ensure that the lamp intensity and the corresponding shape of any projected edge profiles will remain constant throughout the calibration or measurement process.

It was found through re-calibrating the system on a number of occasions during system development, that results were more reliable if the projector was turned on for at least ten to fifteen minutes prior to calibration, and prior to applying it to generate any test images. In fact, due to some of the apparent variability in the projector's output, best results could be obtained through calibrating the system and then taking sets of application images in one continuous operation without turning off the projector.

It should be made clear that no variations or artefacts that were readily visible to the human eye were detected in the projected images. However, through a number of iterations of system calibration and use, it became apparent that the projector introduced some measurable degree of variability if precautions regarding warm-up time were not taken.

5.6.2 Digital Light Projection (DLP) System Technology

The Plus U2 data projector uses DLP™ technology [2] based on an optical semiconductor called a Digital Micromirror Device (DMD) chip invented in 1987 by Texas Instruments [1]. The DMD is basically an extremely precise light switch that enables light to be modulated digitally.

A DMD chip has on its surface several hundred thousand microscopic mirrors – spaced less than 1 micron apart – and arranged in a rectangular array such that the mirrors correspond to the pixels in the image to be displayed. The mirrors can be individually rotated ±10-12°, to an on or off state (Figure 5.12). In the on state, light from the projector bulb is reflected into the lens making the pixel appear bright on the screen. In the off state, the light is directed elsewhere (usually onto a heatsink), making the pixel appear dark.

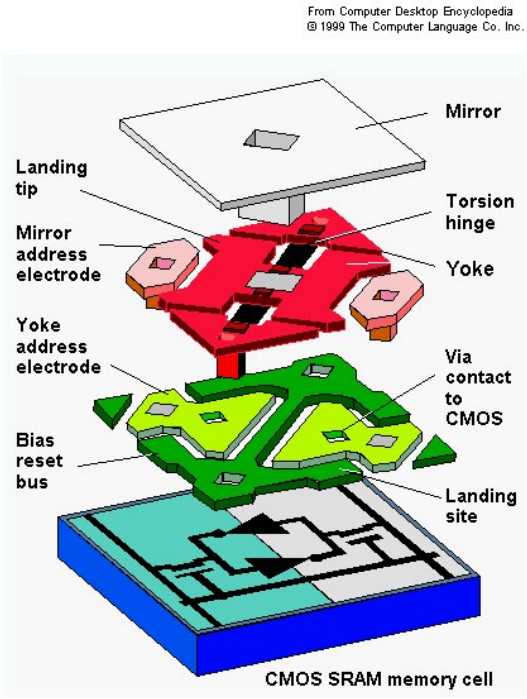


Figure 5.12 Detail of a Single DMD Mirror Pixel Cell [3]

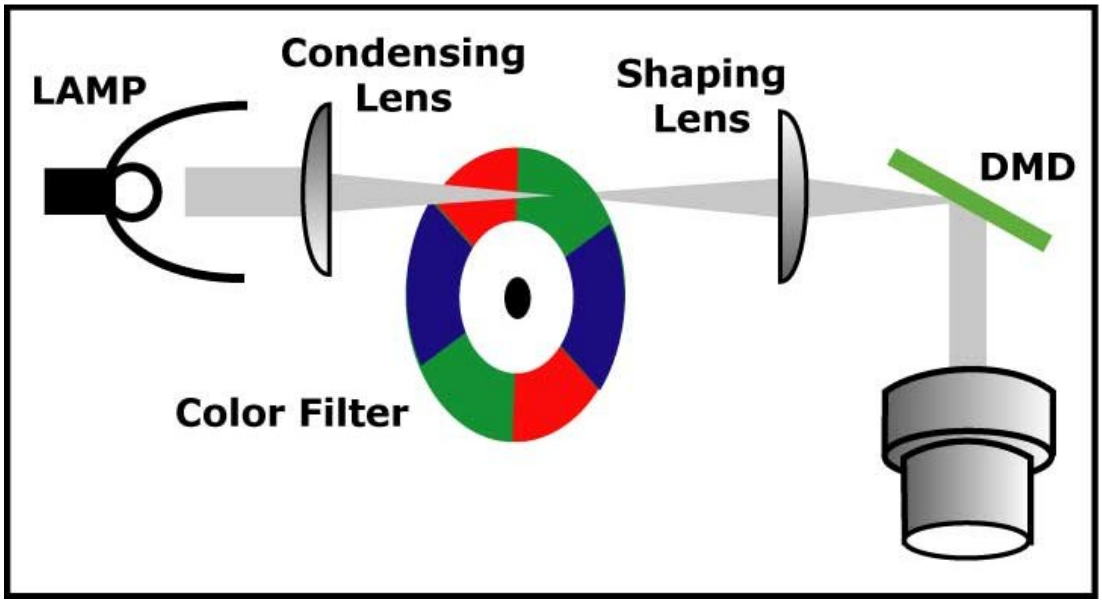


Figure 5.13 The DLP System including a Digital Micromirror Device [2]

These mirrors are literally capable of switching on and off thousands of times per second and are used to direct light towards, and away from, a dedicated pixel space. The duration of the on/off timing determines the level of grey seen in the pixel.

By integrating this greyscale capability with a 6-panel colour wheel (2x RGB), the DLP system is able to produce more than 16 million colours (Figure 5.13). A DMD system can be made up of a single chip or 3 chips, resulting in even greater colour reproduction.

The Plus U2 projector is a single chip DLP device. This is more than adequate for most normal projector applications, but for use in the DfD system, the single-chip arrangement is a likely source of unwanted artefact in the projected image. There are two key sources of potential error.

a) The mirrors used in the DLP system are prone to internal reflections that can – in certain conditions – enter the lens and form part of the image output. It should be noted that the 'keystone' adjustment facility – that was particularly useful in this system arrangement – may also have contributed to this effect.

When using the system in a ‘blacked-out’ laboratory room, it was occasionally possible to visually detect a slight reflection in part of the pattern image.

b) The so-called DLP 'Rainbow Effect' (or RBE) is an artefact unique to single-chip DLP projectors. The artefact appears as a rainbow or multi-colour shimmer briefly noticeable when the viewer looks rapidly from one part of the screen to another. It manifests itself as a secondary image in rainbow colours that appears at the viewer's peripheral vision and is generally noticeable when the viewer shifts their focus from a high contrast to a low contrast area. Though difficult to quantify, it is suspected that one effect of the single-chip DLP system is to produce a slight variance in average light intensity over time – particularly when a greyscale image is being projected (as is the case in the DfD research system). This would have no consequence for normal data projection use, but can cause problems in the DfD system developed.

Either of the above DLP-based error sources could cause some variability in the results obtained throughout the development of this DfD system. Never the less, despite the potential problems described above, the projector provided the means to introduce considerable – and novel – enhancements to the development of the system, as described in the remainder of this chapter.

5.7 Variation in depth estimates along an image row

Another effect that was noted in the depth maps produced by the 2-D look-up table system was a notable tendency for the map to estimate slightly greater depths in the centre of the image rows, and correspondingly slightly smaller depths towards the outer ends of the image rows. Figure 5.14 plots the variation in depth estimate where the data used is the average look-up table index value that corresponds to the depth

estimate. This effect can also be seen in the results of the image averaging exercise (Appendix B-8), although the process of image averaging is not in itself contributable to this effect. Note that a lower look-up table index represents a lower depth, and vice versa. Hence, the look-up table index values plotted in Figure 5.14 will estimate smaller depths towards the outer edges of the row and greater depths towards the centre of the row.

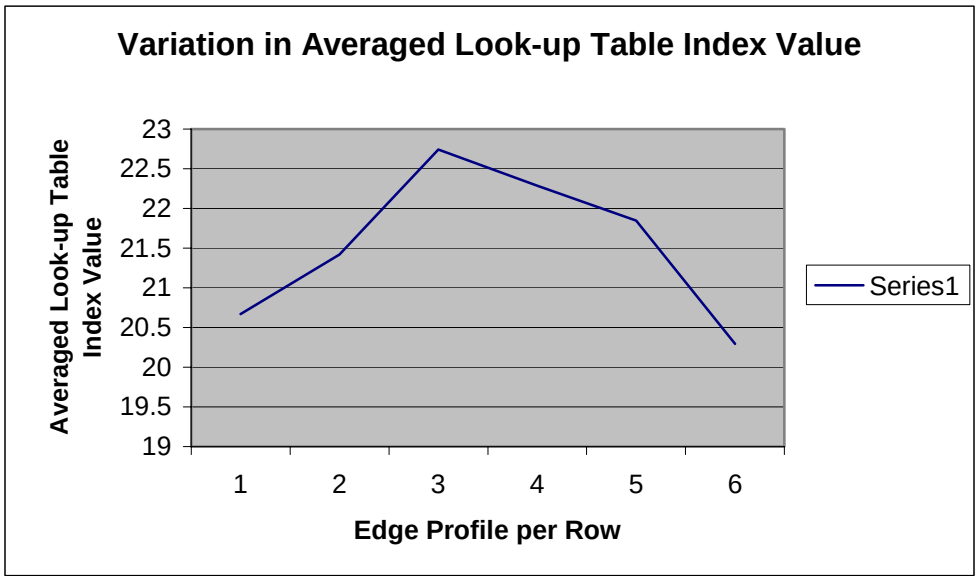


Figure 5.14 Horizontal Variation in Edge Profile Look-up Value

Figure 5.15 shows the general trend of variation in the intensity levels of the image of the projected pattern across a single image row. This was examined as a means of identifying the possible cause of the problem. To continue this investigation further, the profiles of the six central rising edges from the row profile plot of Figure 5.15 were examined individually. The results of this are recorded in Appendix B-9, where it can be seen that each of the six normalised edge profiles examined (left-to-right across the row) produce a look-up value that will generate a greater depth estimate in the centre of the row, and a smaller depth estimate towards the left and right ends of the row. The plot for these edge profile values is repeated below for clarity (Figure 5.16)

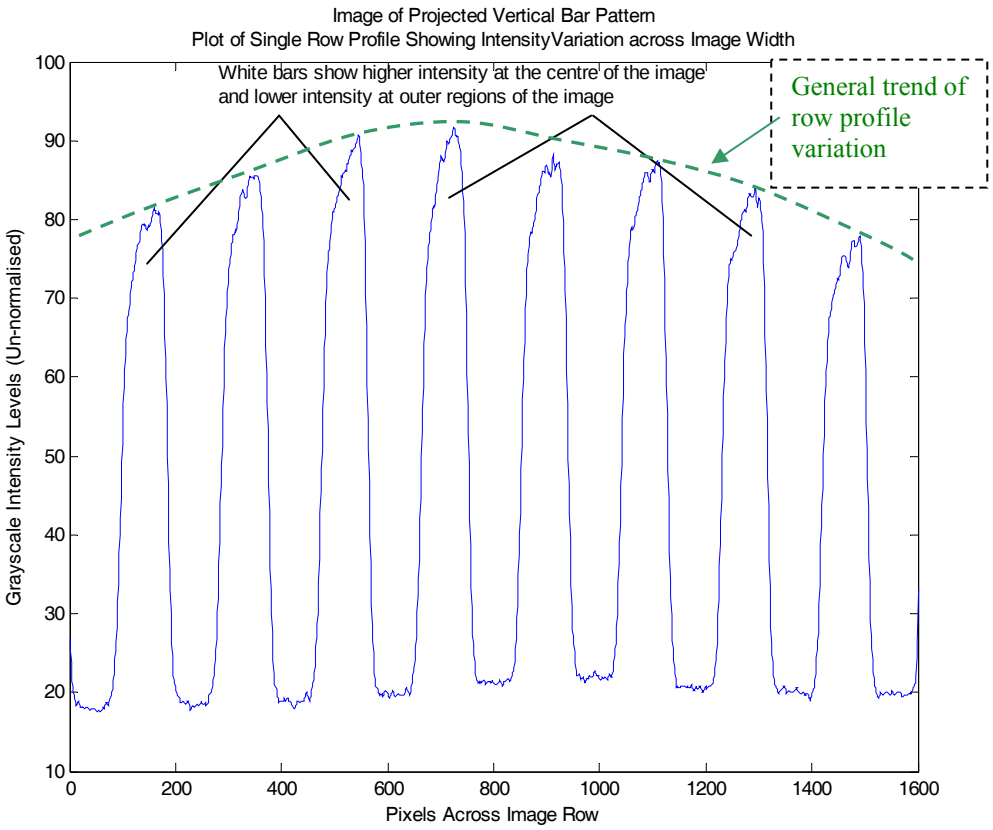


Figure 5.15 Variation in Profile Intensity Across an Image Row

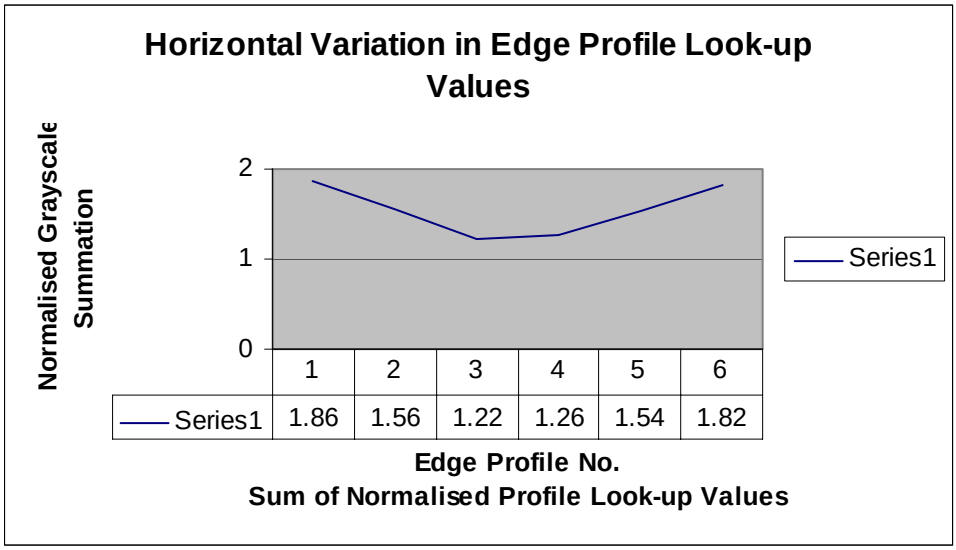


Figure 5.16 Plot of six edge profile look-up values across image row

Figure 5.17 provides an illustration of why this problem is related to the projector/camera configuration. The projector clearly generates a divergent image, the principal rays of which will scatter as they hit the planar screen. The camera lens

that is positioned below the projector lens on the same optical axis effectively views a convergent image of these scattered rays from the planar screen. However, there is also likely to be some reflective component, as illustrated by the principle rays of reflection in Figure 5.17.

It can be seen that, in Figure 5.17, the central region of the image reflects back almost directly to the camera lens causing less distortion to the shape of projected edges within that region. Whereas the principal rays reflecting from regions further away from the vertical image centre do not reflect directly back to the camera lens and the reduced intensity that the camera lens receives from those regions can potentially distort the shape of a projected edge profile at that point. The consistency with which this effect was found, both throughout the length of an individual image, and across a variation of images at different depths, led to the conclusion that this was a systematic error caused by the configuration of the system.

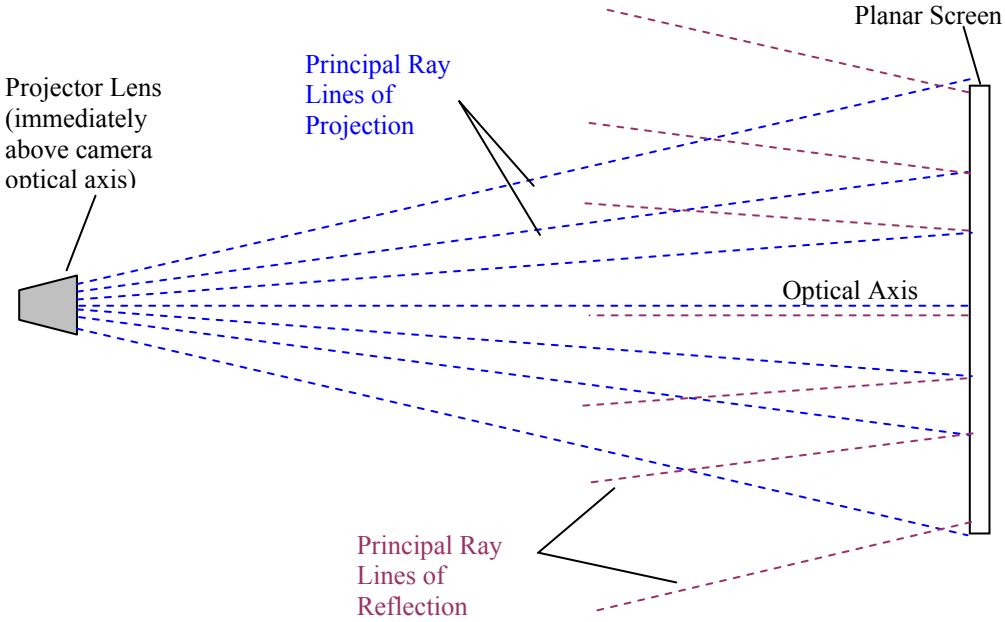


Figure 5.17 Reflection of Projected Pattern from Planar Screen

5.8 Development of a 3-D look-up table to improve the horizontal spatial accuracy of depth estimates

In order to overcome some of the problems of inaccuracy in depth estimation across the image width as discussed in section 5.7 – and to a lesser extent to cater for some of the less predictable inaccuracies caused by DLP technology in the data projector (section 5.6) – it was determined that the DfD algorithm should be modified and improved.

This section explains how the DfD system was expanded to work in 3-dimensions rather than two. The purpose of this expansion was to produce 'standard' edge profiles not only for each actual depth interval but also for intervals across the row of each depth image. Figure 5.18 illustrates the principle of the 3-D arrangement.

When the 3-D look-up table is being created, an image is taken of the structured light pattern at each depth interval (e.g. 90 cm to 114 cm in 1 cm steps). Each depth image is then analysed by the same method described in section 5.1. However, rather than producing a 'standard' edge profile to represent the entire image at that depth, a standard profile is produced for each valid profile across the image (i.e. P1, P2, P3, etc). So where the image of the projected light pattern produces e.g. six valid edge profiles per row, the 3-dimensional look-up table size will be 100 x 6 x 25, where 100 is the length of the normalised profile, 6 is the number of profiles per depth image, and 25 is the number of depth images (***I***).

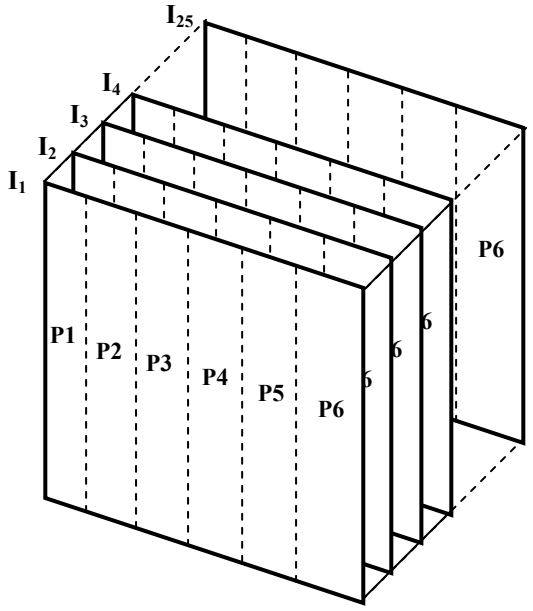


Figure 5.18 3-D Look-up Table

The algorithm that uses this 3-D look-up table to analyse an image of an object under investigation, will follow the steps given below in order to make each depth estimate.

1. Examine each row and extract each valid edge profile of the projected light pattern. Note the row number and pixel position across the row at which the edge is detected.
2. Normalise the edge profile vector
3. For the extracted profile position, e.g. P1, compare the look-up value ('x' range 30 to 50 pixels) with the equivalent look-up value for all P1 profiles in depths I_1 to I_{25} until the best match is found, and hence the depth estimate is made.
4. Place the depth estimate in the output 1200 x 1600 depth map using row and pixel locations noted in step 1.

As the individual depth image size is 1200 x 1600 (row x column) pixels, it can be seen that this 3-D form of look-up table can potentially generate 1200 x 6 = 7200 depth estimates per image.

5.8.1 Increasing the spatial resolution of depth estimates by moving the projected structured light pattern

Identifying the horizontal position of an edge and matching it to a standard profile that was derived from that horizontal position for each depth, made the DfD system more robust in terms of reduced variability in depth estimation over the image width. However, there was still a problem of poor spatial resolution of the depth estimates within the depth map. It was possible to implement the algorithm on every image row, and hence produce a very useful vertical depth map resolution of 1/1200. In the case of horizontal resolution of the depth map, this was heavily constrained by the number of edges that could be extracted from the structured light pattern (in this case giving a horizontal depth estimate resolution of only 1/6).

Two possible methods of improving the horizontal depth resolution were identified.

a. Reduce the width of the black/white vertical bars in the projected structured light pattern in order to create more vertical edges per image row.
b. Use the flexibility of the data projector to move the projected pattern horizontally across the image scene. This would allow the edges to be repositioned and effectively, re-used, to create more edge profiles per row.

Detail of the feasibility of these two options is given below.

a) Modifying the size of the projected structured light pattern

The projected structured light pattern was produced via a MATLAB program running on a laptop computer, and connected to the data projector. In theory the width of each of the vertical black/white pattern bars could be as small as one element of the 480 x 640 matrix used to generate the pattern on the laptop computer (see Appendix A-1). In practice, by the time this matrix pattern is projected onto the scene and is then captured by the 1200 x 1600 UEye camera, a black-to-white edge of the pattern would still correspond to a number of image pixels that would form the edge profile. So in one sense, that represents the smallest size of pattern (and hence, maximum number of vertical edges per row that could be used).

However, it is the shape of this edge profile that becomes problematic if the width of the black/white vertical bars is made too narrow in the projected image.

Consider Figure 5.19.

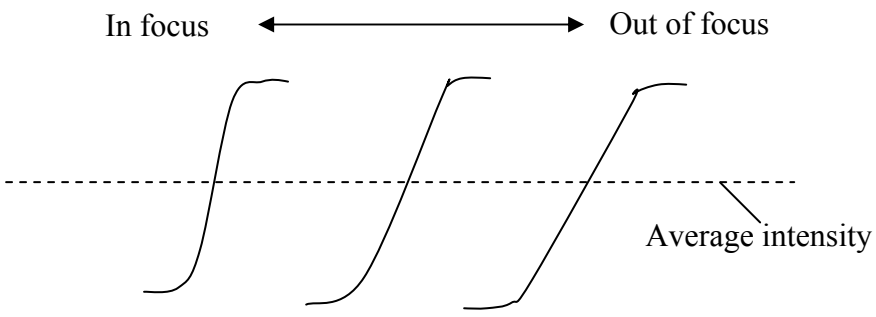


Figure 5.19 Typical variation in edge profile shape as degree of edge focus changes

Figure 5.19 shows that – viewing from left to right – as the black-to-white edge becomes more out-of-focus the edge profile shape changes. The profile shapes in Figure 5.19 are simplified representations, but it can be seen that variation of the characteristic elongated 'S' shape (see also, Figure 4.16 Analysis of Optimum Depth Cue Features) is an important feature of the depth cue that was chosen in Chapter 4. Therefore, it is important that the edge profiles retain this characteristic shape throught the working defocus range for which the system is designed.

Figure 5.20 illustrates the basic form that a row profile of an image of the projected light pattern should retain.

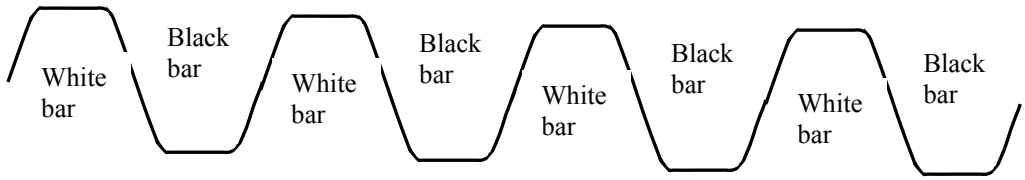


Figure 5.20 Typical row profile from planar pattern image

However, Figure 5.21 illustrates what happens if the projected vertical bar pattern is made too small. If the vertical bars are too close together as the edges begin to defocus, the blur effect of the light energy present in the white bars rapidly causes the white bars to merge into each other across the black bars. Effectively, in terms of the row profile of pixel intensities in the captured image, the pattern forms something closer to a sine wave as shown in Figure 5.21. It can be seen that only the fundamental frequency of the input square wave remains.

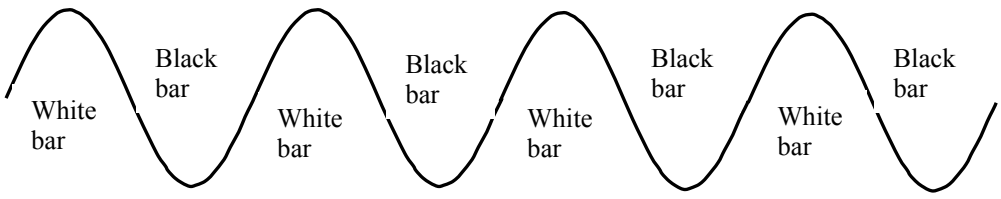


Figure 5.21 Typical row profile from defocused pattern of insufficient bar width size

Note that the maximum and minimum intensity levels that characterise the peaks of the black and white bars will both decrease in magnitude as defocus increases, and the essential shape of the black-to-white edge profile on which the DfD algorithm depends is effectively lost.

The optimum size of pattern for projection was determined empirically by modifying the width of the black and white bars, projecting the pattern onto a planar screen, and taking images at each depth. These images were then analysed to determine whether the row profile retained the form of Figure 5.20 throughout the depth range used.

There was an obvious balance between the minimum size of pattern and the depth range for which the system was designed. It was eventually determined that a pattern period of 20 pixels (i.e. 10 pixels each for both black and white vertical bar width) would be used in the pattern generating program. This would enable 10 valid edges per row to be extracted from the UEye camera image, and would increase the potential number of depth estimates in the depth map from 7,200 to 12,000 (10 x 1200). In order to constrain the depth map size in terms of memory requirement, the working depth range was reduced to 16 cm (actual range 90 cm to 106 cm inclusive – i.e. 17 depth images used to generate depth map). Figure 5.22 shows the format of the revised 3-D depth map.

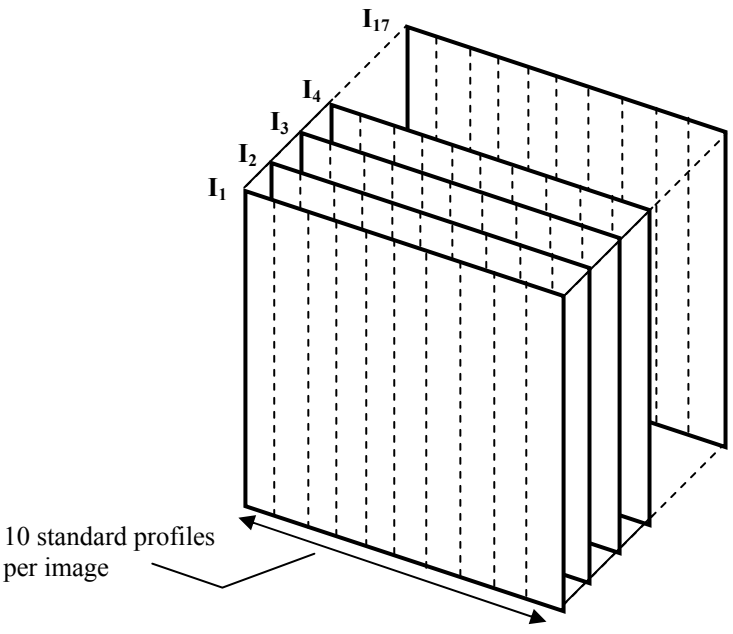


Figure 5.22 Revised depth map for optimum pattern size

b) Moving the projected pattern across the object scene

Having established the optimum size of the structured light pattern and the associated depth range, a novel feature that would substantially increase the spatial resolution of depth estimates was devised. This was specifically enabled by the ability of the data projector to move the pattern position during the image capture operation.

As the matrix used to generate the structured light pattern now had a period of 20 elements (effectively 20 pattern image pixels) in the horizontal direction, it was possible to sequentially move the pattern (left-to-right horizontally) 20 times before it returned to the same horizontal position.

From the point of view of the camera image, this meant that each valid rising edge in the row profile could effectively be 're-used' as it moved through 20 equally spaced horizontal positions. Figure 5.23 shows the type of image that would be seen by the UEye camera when the optimised light pattern is projected onto a planar screen. The small arrows illustrate the positions of the ten rising edges available for depth estimation when the projected pattern is in its initial position.

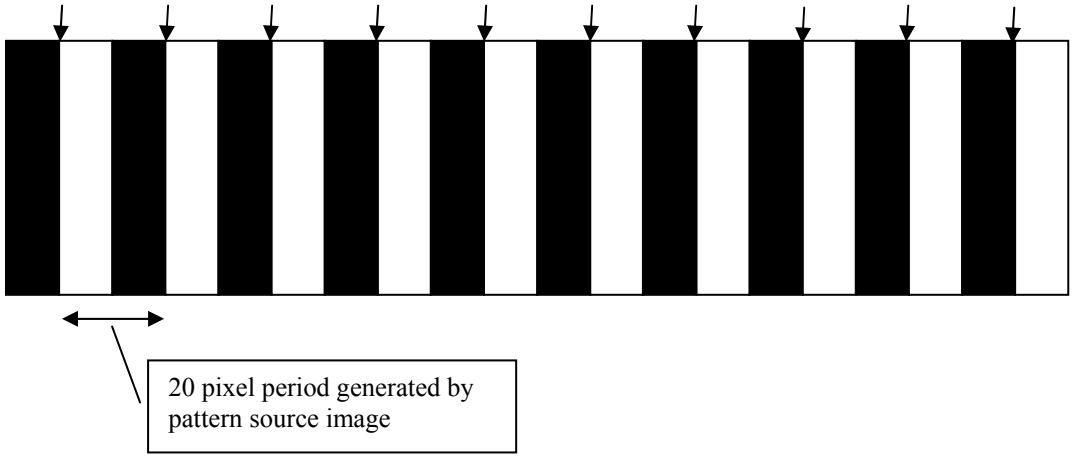


Figure 5.23 Available rising edges from optimised pattern

Left-to-right movement of the pattern by one source image pixel corresponded to each edge position moving approximately five pixels in the UEye camera image.

By being able to perform twenty right-shift operations on the pattern, the number of valid edges and hence depth estimations that could be made per row was now increased to 20 x 10 = 200. Hence the 1200 x 1600 UEye image could now produce a total of 1200 x 200 = 240,000 depth estimates.

This idea was entirely novel, and conceptually relatively simple. However, it was rather more difficult to achieve in practice than the description of the idea may suggest. The method required to implement this concept and to generate a larger 3-D look-up table, involved the following steps;

1. Set the planar screen to the first depth in the range (e.g. 90 cm).
2. Project the optimised structured light pattern in its initial position.
3. Capture twenty images of the pattern at that depth with the pattern shifted-right by one source image pixel for each of the twenty images.
4. Sort the twenty images into their correct left-to-right order then extract the 200 edge profiles also in correct left-to-right order (see explanation below).
5. Analyse each edge profile and place it in its correct position in the 1200 x 1600 depth map.
6. Repeat for each depth (i.e. 17 x 20 images)

Step 4 of the above algorithm requires more explanation. It may initially appear that – if the pattern is moved twenty times from left-to-right, and an image is taken of each pattern position – then those images are already in correct left-to-right order.

However, section 4.6.2 of chapter 4 described how the algorithm to extract valid edge profiles from each row was constructed. It showed that – in order to avoid selecting pattern edges that were too close to the image edge, and hence may provide an incomplete profile – the algorithm would always detect the second edge in from either end of the row. This means that it is necessary to determine which of the

twenty images has the left-most valid edge, and then determine which has the next left-most valid edge, etc. Figure 5.24 illustrates this requirement.

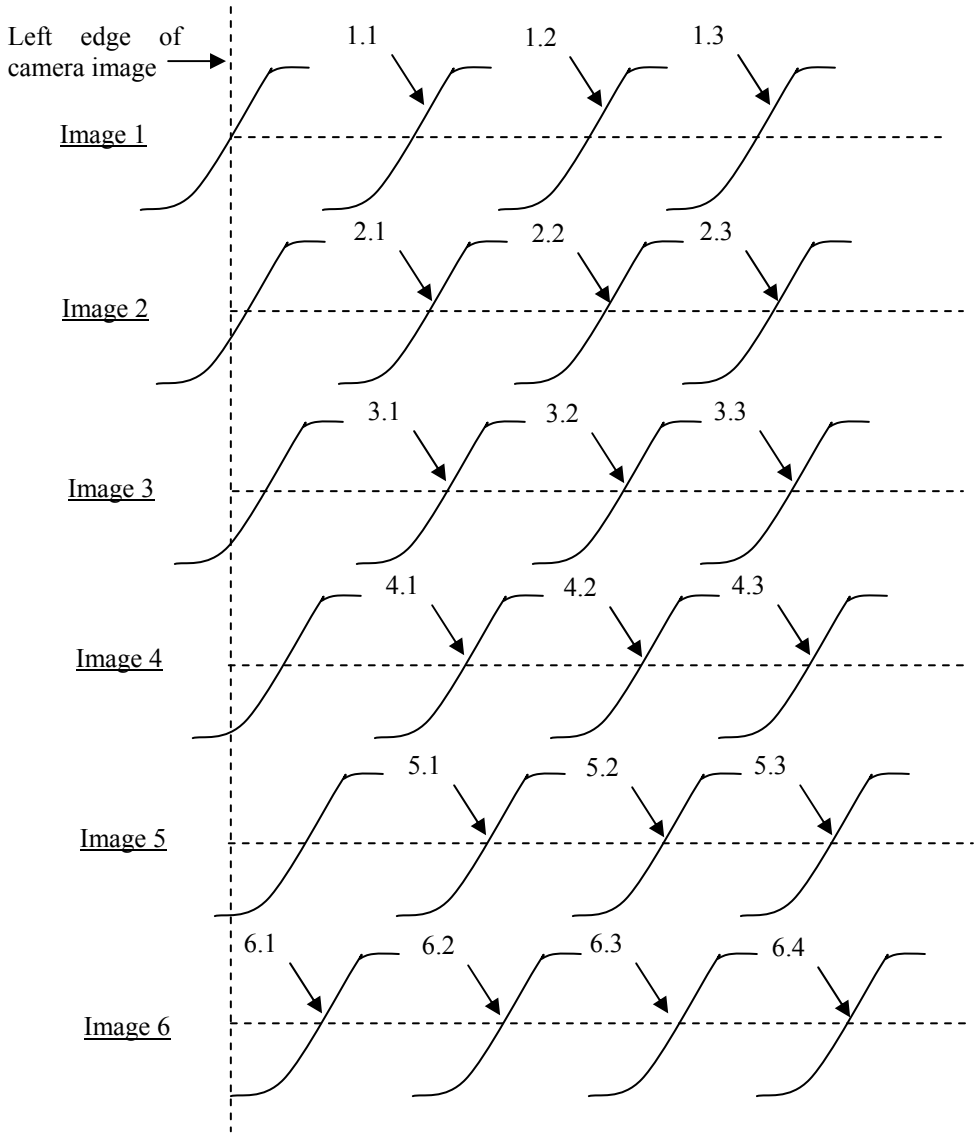


Figure 5.24 Alignment of Valid Edges on Pattern-Shifted Images

Figure 5.24 illustrates why image 1 of the twenty images taken per depth is not necessarily taken as the left-most image. It can be seen that the first valid edge profile on an image 1 row (i.e. edge 1.1) is actually the second edge along the row. As the pattern is moved left-to-right on subsequent images, the non-valid edge profile on the far left of the row eventually – by image 6 – becomes valid. The horizontal position of edge profile 6.1 is now clearly the left-most valid profile when

compared to 1.1, 2.1, etc. Further right-shift movements of the pattern until all twenty shifts (and all 20 images) are complete, should not cause this effect to reoccur again because the twenty shifts occur within one period of the horizontal pattern period. Hence, from the example of Figure 5.24, the sequence in which the extracted edge profiles would be mapped horizontally would be,

6.1, 7.1, 8.1, 9.1, 10.1, 11.1, 12.1, 13.1, 14.1, 15.1, 16.1, 17.1, 18.1, 19.1, 20.1, 1.1, 2.1, 3.1, 4.1, 5.1, 6.2, 7.2, 8.2, ……. 20.10, 1.10, 2.10, 3.10, 4.10, 5.10.

Notice that the correct left-to-right sequence of all first valid edge profiles is 6 through to 20, back to 1 through to 5. This then repeats for all second valid profiles through to the tenth valid profile on all images, giving 200 profiles in correct left-to-right order.

This adds computational complexity, as the 20 images per depth have to be put into correct left-to-right order before the analysis of each image can begin. For generation of the 3-D look-up table, this requires a store of twenty 1200x1600 images per depth, and an edge sequence vector (similar to that shown above) to be generated. To generate the 'standard' profile for the $n/200^{th}$ column, each valid edge will be averaged with its equivalent valid edge on all rows in the same image. Fortunately, repeating this for all 200 valid edges on each of the 17 depth images is only required in order to generate the depth map. So the time overhead is not an issue here.

When applying this method to produce the depth map of an object, it would still be required to capture a single set of 20 images of the right-shifted pattern. However, as each valid edge profile is identified and extracted, its horizontal position would determine which column of the 3-D look-up table the edge profile would be matched with. Effectively, by reference to the illustration of the 3-D look-up table in Figure

5.25, this is like looking through one vector of column values from I1 through to I17 until the best match is found, and hence the best depth estimation is made.

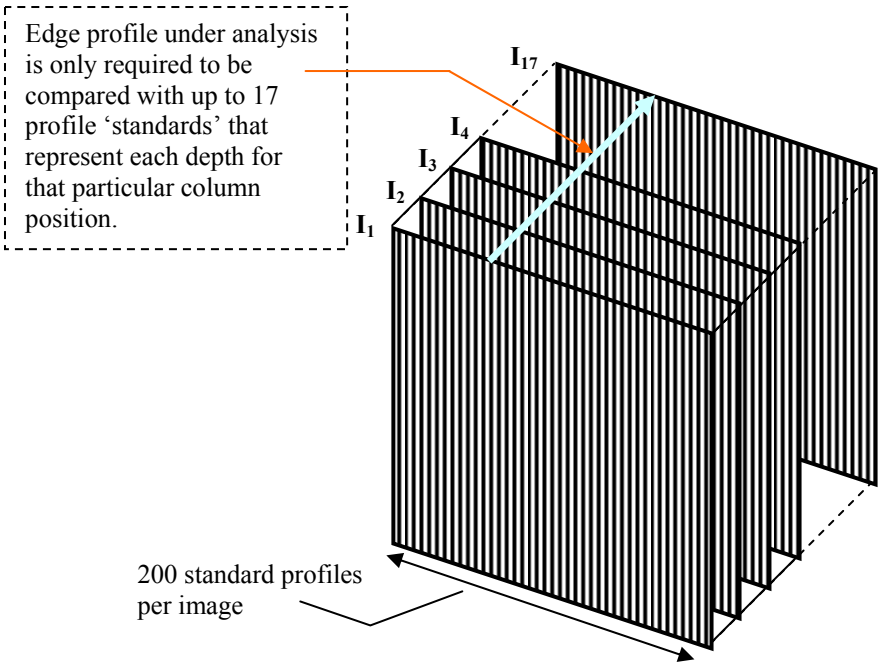


Figure 5.25 Application of high-resolution 3-D depth map

5.8.2 Improvement in horizontal variation of depth estimates by using the 3-D look-up table method

The 3-D look-up table was generated by using the method described in 5.8.1 above. Twenty images of the right-shifted projected pattern were taken at all depths in the range 90 cm to 106 cm inclusive (i.e. 17 depths at 1 cm interval), and the look-up table of size 1200 x 200 x 100 (no. of rows **x** no. of edge profiles **x** length of normalised profile) was generated.

In order to test the improvement in accuracy with respect to reduced variance of depth estimates across the width of a test image, the following experiment (Experiment 8) was carried out.

5.9 Experiment 8: Evaluating the effectiveness of the 3-D look-up table system.

The purpose of this experiment was to evaluate whether – and to what degree – the application of the 3-D look-up table system had reduced the amount of variability in depth estimation across the width of a test image. The 3-D system had been calibrated as described in section 5.8.2.

The problem of horizontal variability in depth estimation was originally detected in the exercise of section 5.7 where 16 time-lapsed images were averaged (App B-8). The actual depth of the planar screen from which the 16 images were taken was 111 cm from the camera lens, and this depth corresponded to a look-up table index position of 22. Hence, for comparison purposes, the error in depth estimate variation across the width of the image, as recorded in Appendix B-8, can be calculated as,

$$\text{For depth over-estimate: } \left(\frac{22.7405 - 22.0000}{22.0000}\right) \times 100 = 3.37\% \qquad \{5.1\}$$

$$\text{For depth under-estimate: } \left(\frac{20.2942 - 22.0000}{22.0000}\right) \times 100 = -7.75\% \qquad \{5.2\}$$

A similar process was adopted to test the modified system by applying the 3-D look-up table method to a planar screen at a distance of 100 cm from the camera lens. This distance corresponds to a look-up table index of 11. Note that the previous depth of 111 cm was no longer an option as the maximum depth was reduced to 106 cm when the 3-D look-up table system was produced.

The 3-D look-up table system has 200 columns derived from the 200 edge profiles per image row for each of 1200 rows, and the data used for comparison was produced by column averaging.

Appendix B-10 records the average depth estimate in terms of look-up table index value for each of the 200 profile columns in the depth map produced by the 3-D look-up table system. Figure 5.26 shows a plot of the data table in Appendix B-10. The maximum average estimate recorded is 11.1, and the minimum is 10.9.

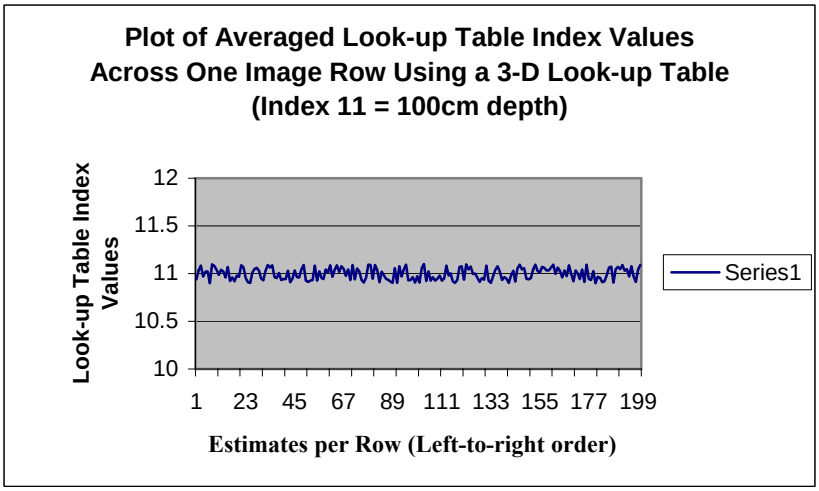


Figure 5.26 Depth Estimate Values Across Image Row using 3-D Look-up Table

Hence, by comparison with the previous error values from equations {5.1} and {5.2} the 3-D system is now producing error values for depth estimate variation across the image width of,

For depth over-estimate: $\left(\frac{11.0998-11.0000}{11.0000}\right)\times 100 = 0.91\%$ {5.3}

For depth under-estimate: $\left(\frac{11.0000-10.9000}{11.0000}\right)\times 100 = -0.91\%$ {5.4}

Hence, a significant improvement in the reduction of horizontal variability of the depth estimates across the depth map width has been achieved. From a depth variation of –7.75% to +3.37%, the system now has a variability of less than ±1%.

5.10 Development of a 4-D look-up table to improve the vertical spatial accuracy of depth estimates.

The 3-D algorithm and its associated look-up table (illustrated in Figure 5.25), was seen to produce significant improvements in terms of reduced variability of depth estimates across the width of the depth map. By modifying the calibration process to produce 200 'standard' edge profiles, each representing their respective 'column' position, any potential sources of such error introduced by system design features are largely calibrated out.

A normal progression to this development is to then consider whether there is variability in depth estimation over the vertical aspect of the generated depth map. Given that the system is calibrated (i.e. the look-up table is generated) by using planar screens that are perpendicular to the optical axis in both 'x' and 'y' directions, then any such variability of depth estimation in the vertical plane would be due to some characteristic of the system design or apparatus (see sections 5.6 and 5.7).

Figure 5.27 shows a plot of depth estimates for six sample edge profiles that represent a selection of edge profiles across the width of the depth map. The plot shows the variation in depth estimate value as each profile is sampled vertically from row 1 through to row 1200. Note that Data 1 is the left-most profile, and Data 6 the right-most profile. It should also be noted that these data were recorded prior to the application of the 3-D look-up table algorithm described above, hence the variability between the edge profiles themselves is greater than it would have been had the 3-D algorithm been applied first.

It can be seen in Figure 5.27 that the more central profiles (Data 3 and Data 4) display less variability with respect to the 'target' look-up table index value of 4 (illustrated by dotted line) than do the outer profiles of Data 1 and Data 6. In the

worst case, Data 1 and Data 2 profiles are over estimating the depth by approximately 1.5 cm at the bottom of the depth map (row 1200).

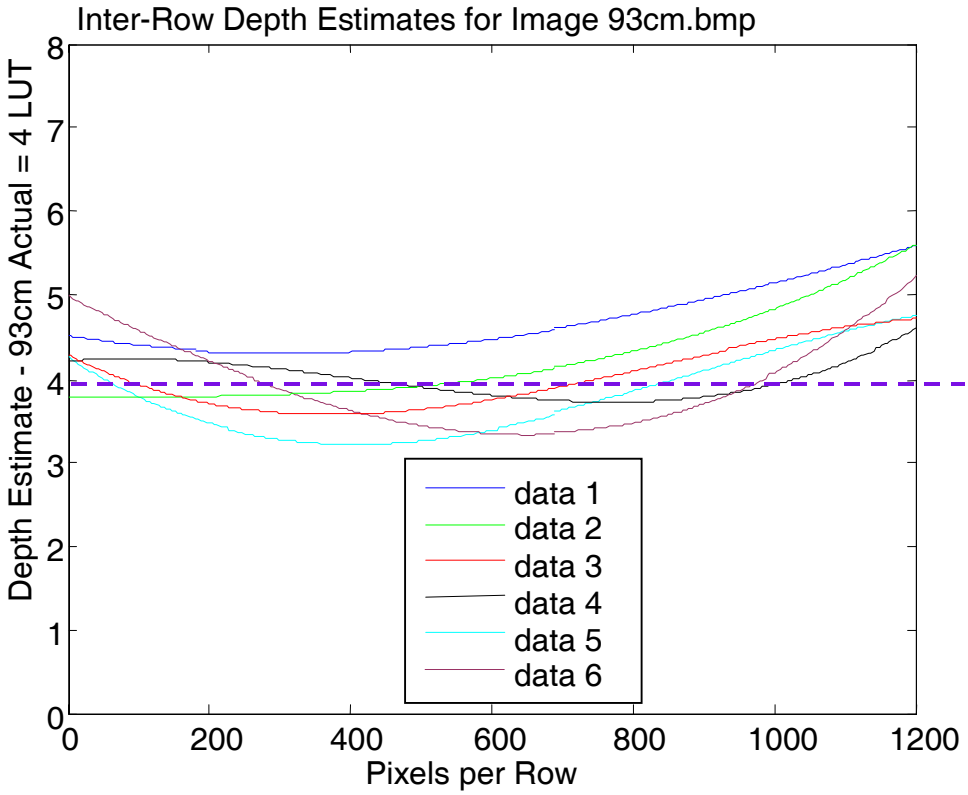


Figure 5.27 Vertical Variation in Depth Estimation

It was determined that – in order to solve this problem – a similar approach to that which solved the problem of horizontal variability should be adopted. Effectively this required the development of an algorithm that would add a fourth dimension to the look-up table system. Rows on each image were arranged in neighbouring groups such that each of the 200 image columns was now also divided vertically, and a 'standard' profile was produced for each vertical group in each column. Hence, each individual depth plane contained a 2-D matrix of depth estimates, and vertical variability was calibrated out.

In developing the program to produce a 4-D look-up table, and an associated program that would use the 4-D table to analyse object images, it was initially determined that a vertical interval of 200 rows would be used. Given that the size of

the depth map is equivalent to the size of the object image, i.e. 1200 x 1600, then this effectively produces an analysis window of size 6 x 8 pixels (1200/200 x 1600/200). That is, for each 6 x 8 pixel window across the image there is a standard edge profile for each depth in the depth range (currently 90 cm to 106 cm – index 1 to 17).

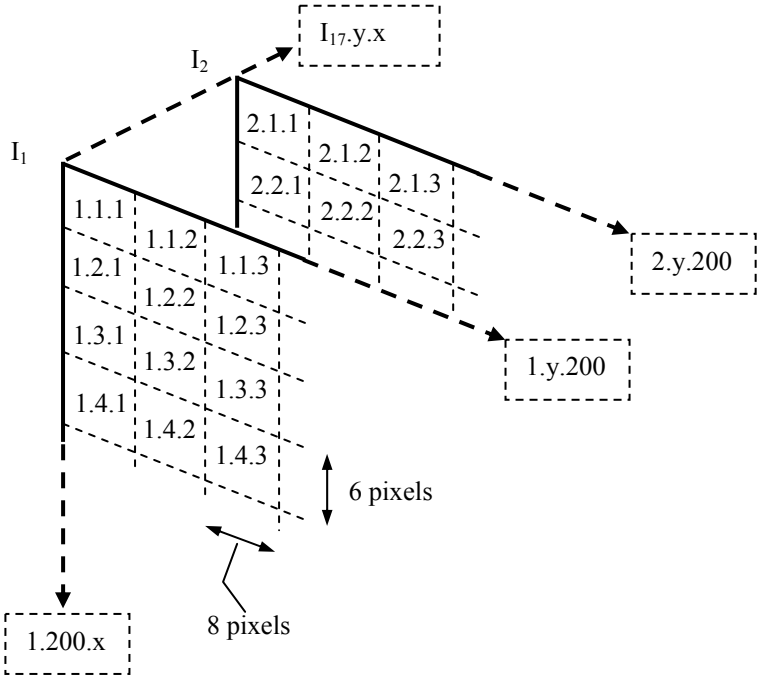


Figure 5.28 Layout of 4-D Look-up Table

Figure 5.28 illustrates the layout of the 4-D look-up table, and effectively shows what the four dimensions are (the 1st dimension being the profile vector that each window in the look-up table represents).

The program to generate this look-up table can be seen in Appendix A-9, and the associated program that uses the 4-D table to analyse an application image is listed in Appendix A-10.

There is no particular reason why the vertical resolution of the look-up table should be constrained to 200 intervals. It is trivial in programming terms to modify the vertical resolution to any range between 1 and 1200. However, it should not be necessary to have a dedicated look-up profile for anything less than 6 pixel rows, as

the rate of vertical depth estimate variation (Figure 5.27) would not require this level of resolution.

Improvement in the vertical variation of depth estimates was tested by first generating the 4-D look-up table (Appendix A-9). This comprised a similar process to the 3-D look-up table generation method – 20 images of a right-shifted pattern per depth for 17 depths at 1 cm depth intervals. However, with the 4-D method each 20-image set is analysed to produce a standard edge profile for each vertical 'window' of 6 rows (as per Figure 5.28).

The 4-D application algorithm (Appendix A-10) was then applied to a planar screen at a depth of 93 cm (the same as that shown in Figure 5.27 where the problem was identified).

Four of the edge profile positions (No's 25, 75, 125, & 175 of 200 across each row) were logged to record the depth estimates they generated on each of the 1200 rows. The plots of this recorded data are shown in Figure 5.29. Note that the scale of the vertical axis in this plot is not representative of depth estimate value. The plots were simply segregated vertically for comparison. In fact, the variation in depth estimation for all profiles over the 1200 row range was within ± 0.25 cm (±0.27%).

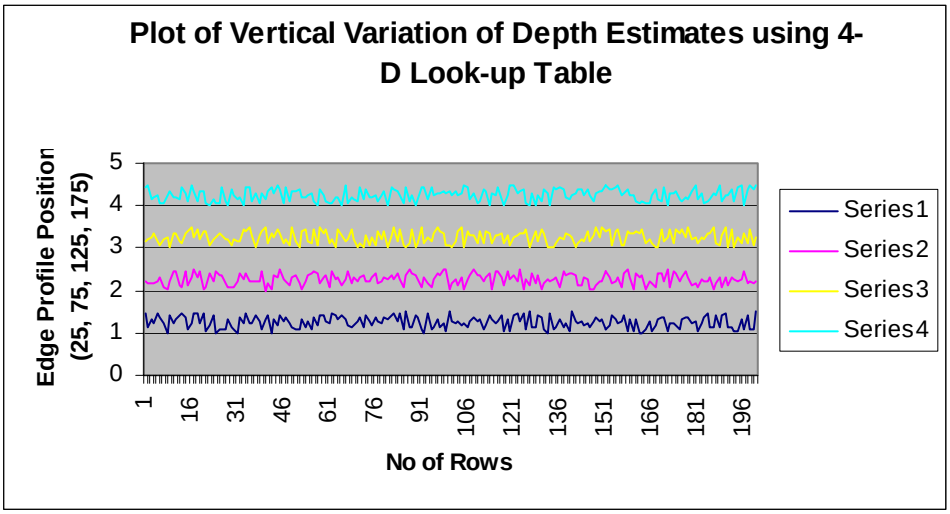


Figure 5.29 Vertical Depth Variation using 4-D Algorithm

This showed significant improvement over the vertical variation error generated by the 3-D algorithm in Figure 5.27. Here the error range varied from approximately –0.5 cm (0.54%) to +1.5 cm (1.66%).

5.11 Testing the performance of the DfD system using the 4-D look-up table algorithm on planar surfaces

The development of the fundamental DfD system was now considered complete. In order to gauge the performance of the system prior to applying it to applications, it was decided to test it on the same type of planar surfaces that had been used to calibrate it (though not on the actual calibration data). Appendix B-11 shows the results for application of the 4-D system on ten depths – 90 cm to 99 cm, at 1 cm intervals. The results in Appendix B-11 include statistical values and a colour-mapped mesh plot for each depth. The mesh plot is a rather more qualitative form of result, but it gives an indication of the consistency of the depth estimate over the area of the planar object. For illustration, the first of these depth results is repeated in Figure 5.30.

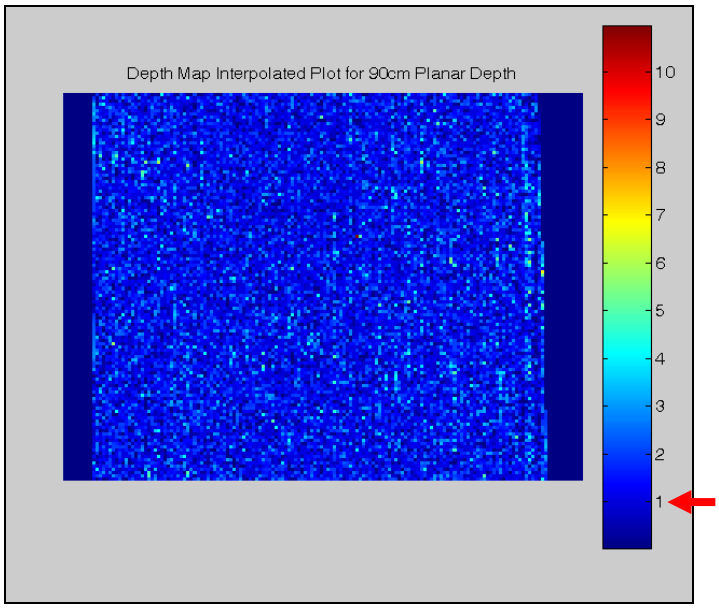


Figure 5.30

Planar Object at 90 cm

The known depth is 90 cm
The mean of calculated depth values = 90.337 cm
The standard deviation of calculated depth values = 1.04
R.M.S. Error = 1.0932
Percentage R.M.S. Error = 1.2147%
Calculated Depth Estimate Error = 0.37449%

Figure 5.30 shows that the estimated depth for the planar object as a whole is 90.337 cm (found by averaging the measured values per depth plane). As the object is actually at 90 cm depth this represents a depth error of 0.375%.

This represents a very good result by comparison to other researchers who have used multiple-image techniques with active illumination methods over a similar depth, e.g. Ghita & Whelan [4] who claim 3.4% error over a distance of 86 cm. It certainly represents a far more robust result than has previously been achieved using a single image technique with active illumination – neither Girod & Scherock [5], nor Hinjosa, et al [6] give any quantitative results of depth estimate from their methods, and only Girod & Scherock [5] display qualitative results of very poor spatial resolution.

The mesh plot in Figure 5.30 shows – qualitatively – that the estimate across the area of the planar surface is generally consistent. The red arrow pointing to the number 1 on the colour-map legend shows that for look-up table index 1 (equivalent to a depth of 90 cm) the mesh should display that colour of blue. The mesh is predominantly of that colour, though there are a number of random points that represent a lighter colour, and hence a greater depth. This is borne out by the calculated standard deviation of the depth estimates being 1.04 cm, and the RMS error of 1.22%.

The developments described in sections 5.9 and 5.10 of this chapter have been generally successful at producing a single-image system capable of generating accurate depth estimates that are broadly consistent over the area of investigation. However, there are random effects generated by the system configuration used in this investigation that will still cause a certain percentage of random error in the results.

The results for the remaining depths show a similar pattern of good levels of average accuracy. A plot of the estimated depth error (%) and RMS error (%) over the 10 cm depth range is shown in Figure 5.31 below.

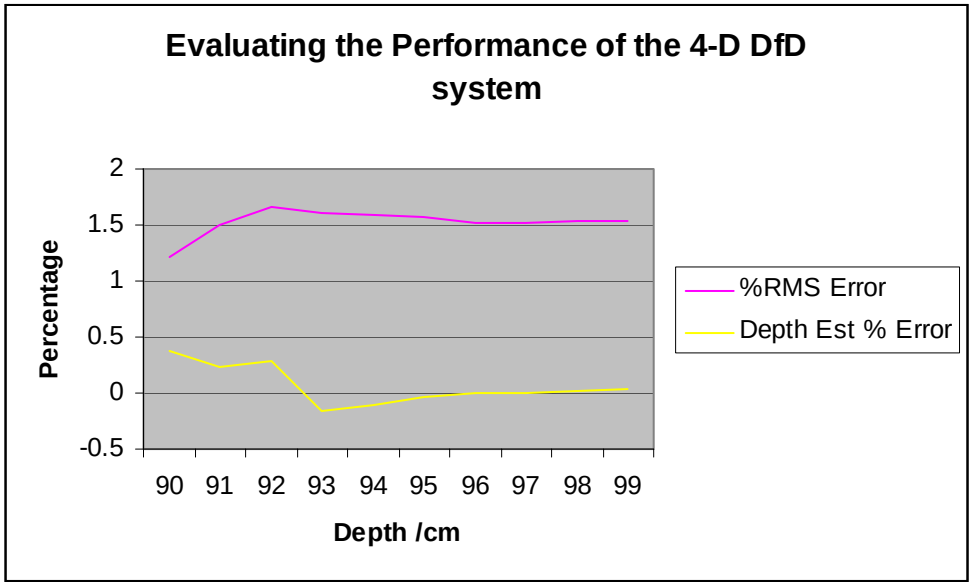


Figure 5.31 Performance of the 4-D DfD system on planar surfaces

The plots of Figure 5.31 show acceptable levels of depth estimate accuracy over the 10 cm range. The percentage RMS error is consistent at approximately 1.5%.

5.12 Neutralising original image texture to improve depth estimate accuracy.

The system developed up to this point had been developed and tested on planar surfaces that were perpendicular to the optical axis. Chapter 6 of this thesis deals with attempts to apply the algorithm to objects that may be neither single-depth nor planar. In this case it was considered possible that the object scene may present sharp edge features and patterned textures that would conflict with the projected structured light edge features being sought by the 4-D DfD system. For this reason it was determined that a function capable of neutralising the texture of the original scene

prior to active illumination could be beneficial to the performance of the system. Effectively, this would mean that the only defocus information remaining in the images being analysed was the variation in edge profile of the structured light pattern. Apart from the potential problem of edge features in the original scene, it was also considered possible that this technique could help to alleviate some of the random error features introduced by the system configuration (e.g. random internal mirror reflections within the data projector that produced unwanted artefacts on the image scene - see section 5.6).

The method adopted to generate a degree of texture neutralisation for application objects can be summarised as follows,

1. Capture an image of the scene illuminated by only the 'white' bar intensity value of the structured light pattern. Dependent upon how the structured pattern is generated, this will be less then or equal to a maximum 255 value in the original pattern matrix.
2. When using the 4-D look-up program to analyse the object scene, load this 'plain' image first and determine its mean intensity value.
3. Generate a 'multiplier' matrix the same size as the image (1200 x 1600) = (M x N). The values in this matrix are determined by the equation,

$$\boldsymbol{M}_{m,n} = \frac{\boldsymbol{mean}}{\boldsymbol{I}_{m,n}} \qquad \{5.5\}$$

where, $M_{m,n}$ = multiplier matrix elements, and $I_{m,n}$ = object image

4. When the images of the object scene illuminated by the structured light pattern are loaded, multiply the image by the 'multiplier' matrix (i.e. array multiplication) to neutralise the texture from the original image,

$$\boldsymbol{I2_{m,n} = I_{m,n} .* M_{m,n}} \qquad \{5.6\}$$

Appendix B-12 gives a qualitative indication of the effect of applying the texture neutralisation algorithm. It shows how the original scene image is used to provide information about texture that is present prior to illumination by the active pattern. It then illustrates how this texture is neutralised by the algorithm, leaving only the texture generated by the structured light pattern itself. Of course, the level of intensity and contrast provided by the structured light pattern entirely dominates the original scene texture, but the removal of original texture helps to improve the accuracy of depth estimation.

Providing direct quantitative evidence of the effect of texture neutralisation on specific image/active illumination features is difficult in terms of showing how it can improve the accuracy of depth analysis. However, Appendix B-13 provides statistical data for the depth accuracy and standard deviation of the application of the 4-D DfD system using texture neutralisation when applied to planar objects at depths 90 cm to 106 cm inclusive. As Appendix B-13 shows, the main purpose of this exercise is to make a comparison of the overall improvement in the performance of the final 4-D system over the original 2-D system, but in order to directly compare the effect of improvement made by the application of texture neutralisation, Figure 5.32 makes a comparison of the results produced in Appendix B-12 (4-D prior to texture neutralisation) and Appendix B-13 (4-D after texture neutralisation).

Figure 5.32 shows that there is a general improvement in depth estimation error over the depth range measured, particularly at the smaller depths where the image is more defocused. However, it is noticeable that the Standard Deviation values have improved significantly over the depth range – typically from approximately 1.5 cm to 1.0 cm. Hence, an improvement in the consistency of depth estimate accuracy over the measured area should be achieved.

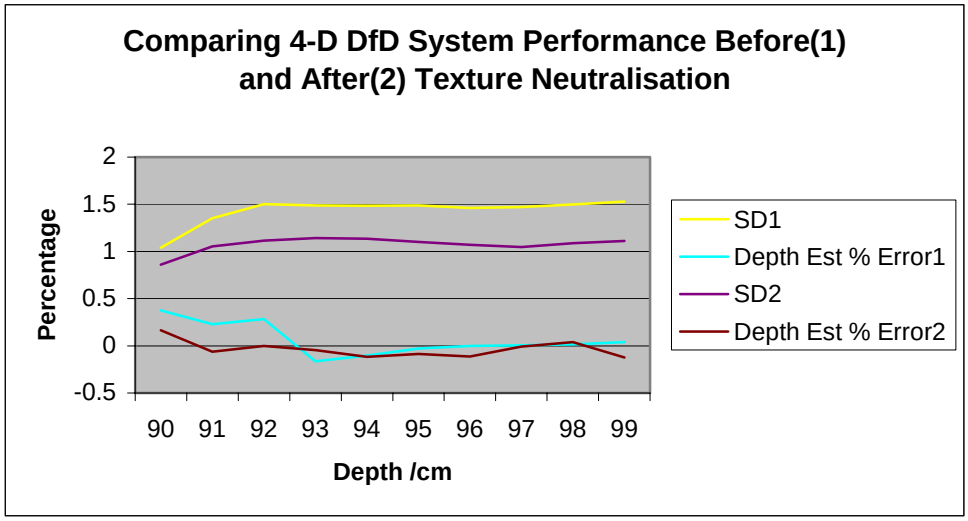


Figure 5.32 Comparison of 4-D DfD Performance Before & After Texture Neutralisation

5.13 Summary of Chapter 5

The principle of using the varying profile of edge features superimposed onto an object scene by projection of a structured light pattern has been developed into a working Depth-from-Defocus system. The DfD system uses a look-up table of 'standard' edge profiles that represent a range of depths. Edge profiles extracted from an object application image are compared to those stored profiles, and a best match provides an estimate of the depth. These depth estimates can then be placed into a depth map from which a plot or mesh can be generated to approximately recreate the 3-dimensional information in the object scene.

The consistency of depth estimate accuracy across the 2-dimensional area of the scene was improved by developing firstly a 3-dimensional form of look-up table, and then finally a 4-dimensional look-up table system. Finally, improvement in the average depth estimate accuracy, and particularly in reducing the standard deviation of the variation in depth estimate accuracy within a given object scene area, was achieved by the inclusion of a texture neutralisation algorithm.

Chapter 5 – References

[1] Digital Micromirror Device - http://www.answers.com/topic/digital-micromirror-device-1 (17-01-2007)

[2] Digital Light Processing (DLP) – Figure 5.11 – http://www.audioholics.com/education/display-formats-technology/display-technologies-guide-lcd-plasma-dlp-lcos-d-ila-crt/display-technologies-guide-lcd-plasma-dlp-lcos-d-ila-crt-page-2 (17-01-2007)

[3] Digital Light Processing (DLP) – Figure 5.10 – http://lookup.computerlanguage.com/host_app/search (13-08-2007)

[4] Ghita O, Whelan P F, "A Video-rate Range Sensor Based on Depth From Defocus", *Opt Laser Technol; v33, No3, pp167-176*, 2001

[5] Girod B, Scherock S, "Depth from Defocus of Structured Light", *SPIE, Optics, Illumination, and Image Sensing for Machine Vision, v1194, pp209-215*, 1989

[6] Hinojosa C, Serrano-Heredia A, Ibarra J, "Recovery of three-dimensional shapes by using defocused structured light", *Optics & Laser Technology; v30, p281-290*, 1998

Chapter 6

Applications of an Active DfD System

This chapter illustrates the results of applying the 4-Dimensional look-up table DfD system – as described in chapter 5 – to a number of 3-D shaped objects.

It should be emphasised that the results shown in Appendix B-13, and the associated plot of those results in Figure 5.32 of chapter 5, are considered to be the definitive set of results for the performance of the system as developed to this point.

The purpose of applying the system to objects of relatively complex 3-D shape was to examine how well the system would cope with features such as edges and curves. To that extent, the following applications – and the mesh plots produced from their generated depth maps – may be considered to be a rather more qualitative than quantitative form of result, though estimation of depth accuracy has been made where possible. The more complex shaped objects represent a rather ambitious and difficult challenge for the prototype system in its present state of development.

6.1 Projecting along the optical axis

In order that the structured pattern was projected onto all aspects of the scene viewed by the camera, it was necessary to modify the physical arrangement of the system to provide projection along the optical axis. This had not previously been a problem when using planar objects, but with objects of more complex 3-D shape there could be areas of the object that would be seen by the camera but that would obscured from receiving the projected pattern from the projector.

The method adopted is illustrated in Figure 6.1 where a half-silvered mirror was used as a beam-splitter in order to direct the light rays from the projector onto the same optical axis as the UEye camera.

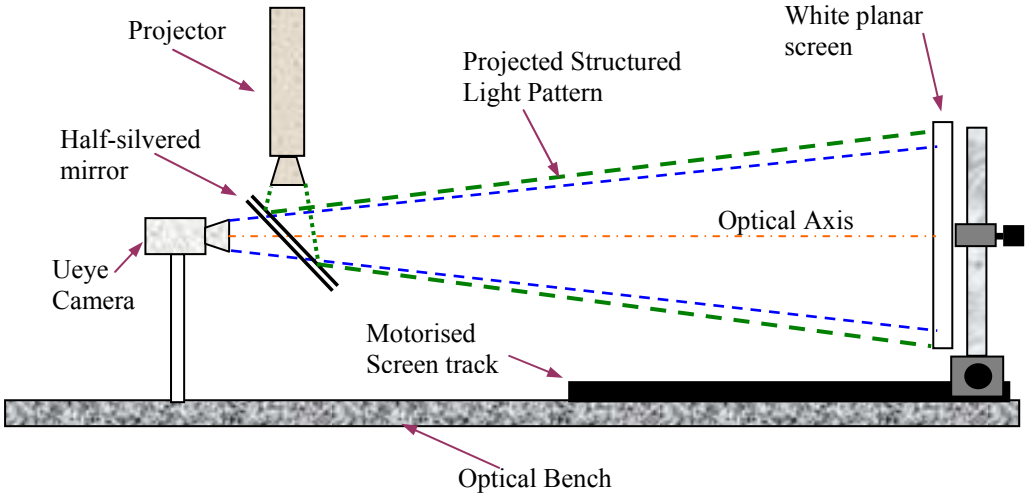


Figure 6.1 Projecting along optical axis

A photograph of the actual half-silvered mirror arrangement can also be seen in Figure 6.2 below.

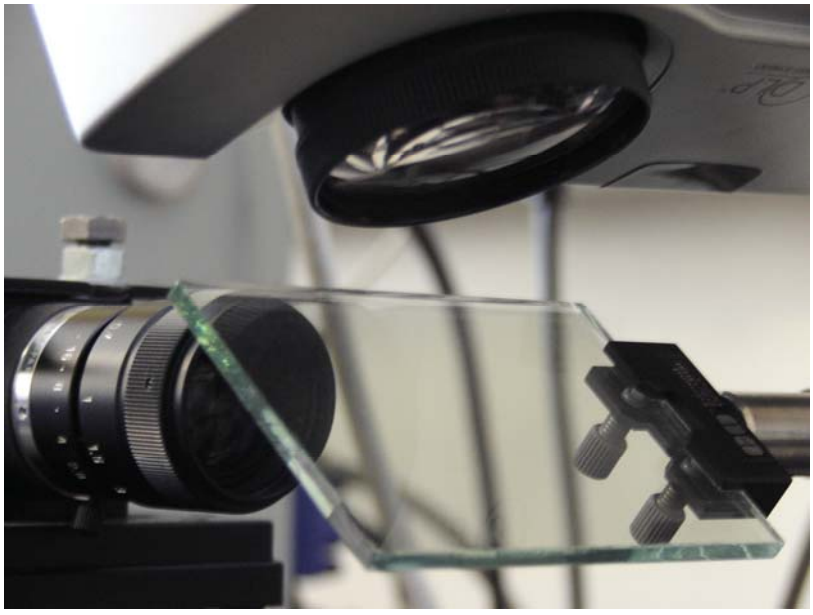


Figure 6.2 Half-silvered mirror arrangement for projecting on optical axis

Naturally, it was necessary to recalibrate the system entirely once the above modifications to the system were completed. This process required the capture of

300 images in order to produce the 4-D look-up table (15 depths with 20 images per depth). Also, as this exercise is undertaken in laboratory conditions (blacked out room, but with the requirement that every depth alteration is accurately measured, and every movement of the projection pattern was completed in correct sequence), it was decided to also take sets of images – within the same laboratory session – for each of the four 3-D objects chosen for investigation. This required the capture of 380 images in total, with the requirement that the depth and image pattern position is precisely known for each image.

In order to make this exercise physically less demanding (and less error prone), a degree of automation was introduced in the form of a computer-controlled, motorised movement for the object mounting/screen fixture. This device used a modified flatbed scanner with associated computer programme and control electronics (courtesy of Dr C Claxton – see 'Acknowledgements'), and can be seen in Figure 6.3 below.

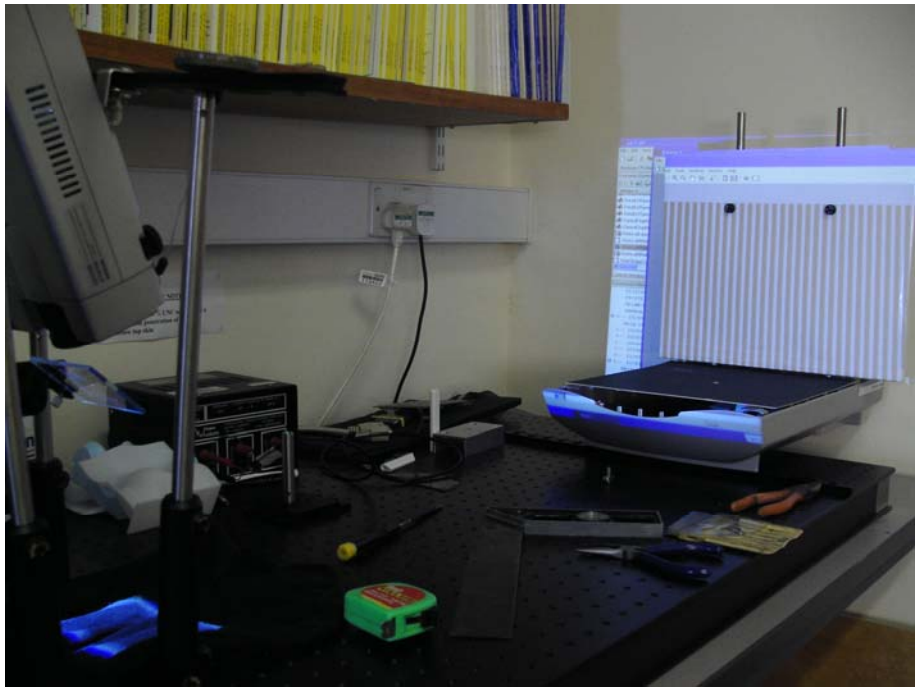


Figure 6.3 Computer-controlled motorised movement carrying object mounting/screen fixture

With this enhancement to the system, the screen was automatically moved a precise distance to the next depth setting where each of twenty projection pattern movements were made, and an image of each pattern position was captured.

6.2 Application to 3-D shaped objects

Following recalibration, the image sets for each of four 3-D shaped objects were captured, and the analysis of these objects is given in the following chapter sections.

6.2.1 Planar shapes at varied depth

This first application was almost a natural progression from the planar screens on which the 4-D system was developed. It used three small white panels offering planar surfaces at three different depths as shown in Figure 6.4.

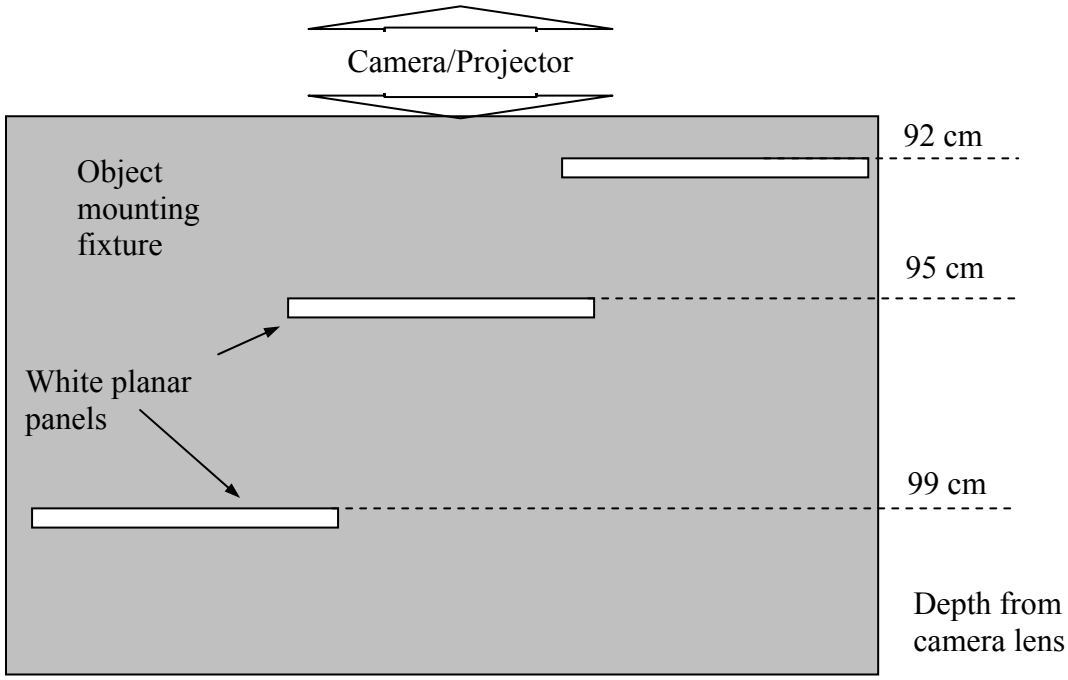


Figure 6.4 Plan view of planar object arrangements

The actual image seen by the camera prior to active illumination can be seen in Figure 6.5. This view is illuminated by the projector at a light intensity level equivalent to that emitted by the 'white' bar of the structured pattern.

Figure 6.5 Image of planar screens with non-active illumination

The image of Figure 6.5 provides some indication of the potential problems that may produce error in the system. Shadowing around the edges of the panels, the suggestion of a shadow in the central lower region of the nearest panel, and part of the object-mounting base that is in view on the lower right corner of the image, all have the potential to produce unwanted effects on the calculated depth map.

The depth map eventually produced by application of the 4-D system is shown in the form of a mesh plot in Figure 6.6. This plot is colour-mapped in order to aid interpretation of the depths shown, and a legend relating depth to colour is also included in the figure. By relating the depths given in Figure 6.4 to those estimated by the system in Figure 6.6, it can be seen that – from a qualitative point of view – the system has performed well. The overall indication of depth for each panel is approximately correct according to the colour scheme used, and the shape – i.e. edges – of the panels are remarkably well defined considering that no function to deal with pattern edge profiles that may be corrupted by being positioned across an object edge has been used.

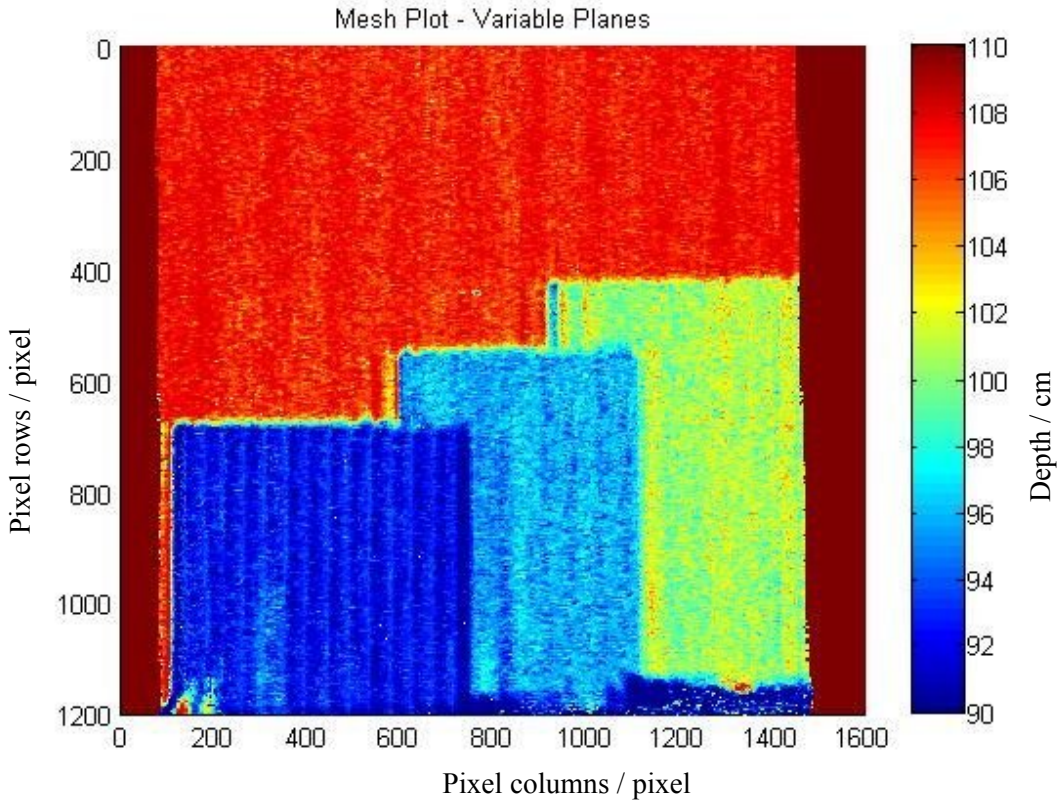


Figure 6.6 Mesh plot of depth map estimates for planar panels at varying depths

Although the plot contains some unwanted artefacts, the texture neutralisation algorithm within the system software appears to have limited the potential effects of shadows, reflections, etc., that would have been present in the original image.

One noticeable effect in Figure 6.6 is the degree of vertical striation that appears to exist within the depth estimates of the near panel at a depth of 92 cm. Fortunately, the colour variation in the striation pattern is not excessive, and broadly remains within the approximate depth range of the panel. However, the periodic effect is consistent, and it will be seen that this effect tends to occur at this depth in other 3-D applications. Further discussion will be made of this effect in Chapter 7 – Conclusions and further work.

In order to provide a more quantitative estimate of the depths measured in this application, a rectangular area of each of the depth planes was extracted from the depth map and analysed (see Figure 6.7).

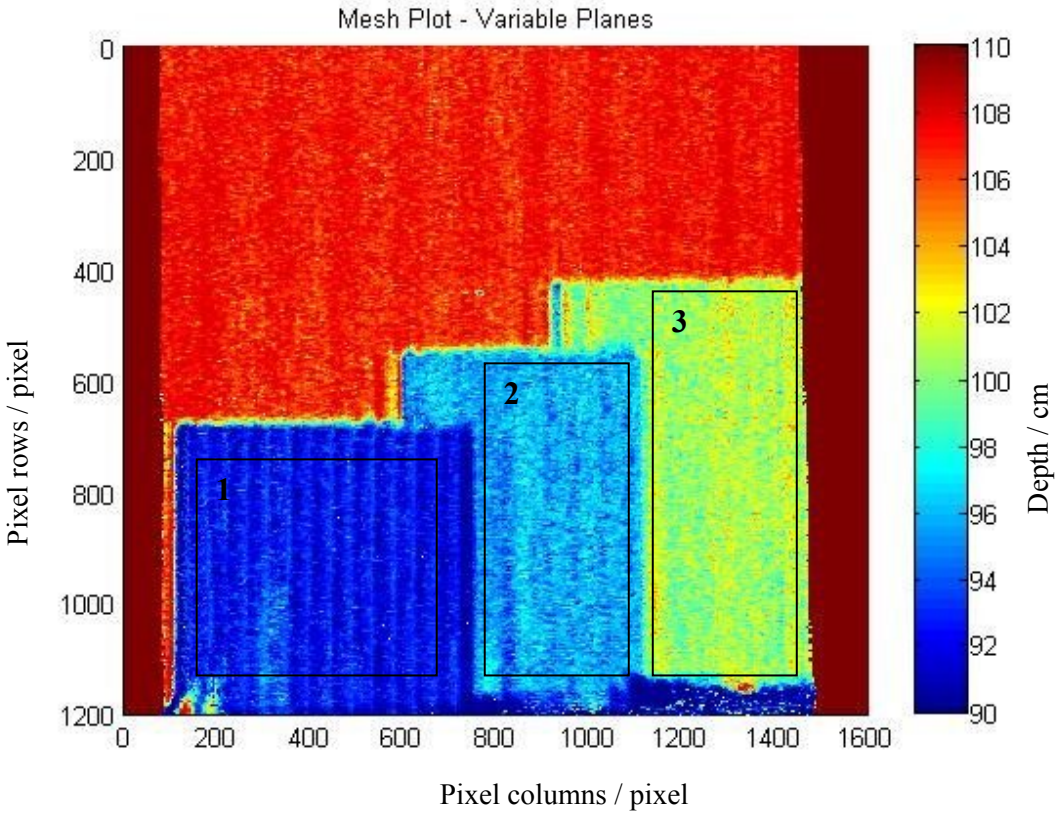


Figure 6.7 Extracted Areas of Depth Planes used for Analysis

The average estimated depth and the variance of measurement for each planar panel area are listed in Table 6.1.

Plane area	**1**	**2**	**3**
Average depth/cm	91.6819	95.0475	99.5150
Variance	0.1252	0.1140	0.0481
Actual depth/cm (see Figure 6.4)	92	95	99

Table 6.1 Depth Map Values for Varied Plane Application

The level of accuracy illustrated by these results is in accordance with the results generally presented for individual depth planes in Chapter 5. Given the slightly

striated pattern of the nearest plane (area 1), the variance for all planes is still relatively low. One positive feature of the depth map plot in Figure 6.6 is that the edges of each plane are quite well defined. It had been considered that some additional process may need to be added in order to deal with the problem of projected pattern edge profiles that were corrupted by being positioned directly over the edge of two planes at different depths. However, the system seems to have coped with this remarkably well, and it can be presumed that the high resolution of the depth map is a major factor in this.

6.2.2 Sloping plane

The basic arrangement for this object can be seen in Figure 6.8 below. The panel support is an open box shape that keeps the panel vertical. It has been shown Figure 6.8 because the open end of the support can be seen by the camera, and hence features in the depth map plot.

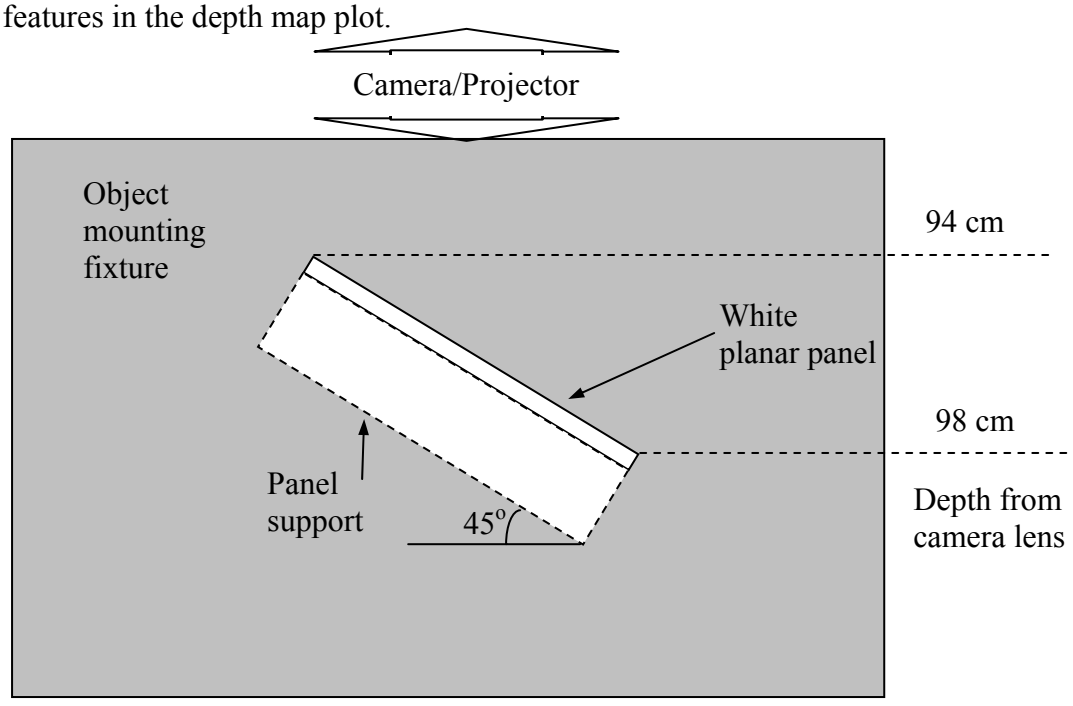


Figure 6.8 Plan view of sloping plane object

The image of the sloping plane under non-active illumination can be seen in Figure 6.9.

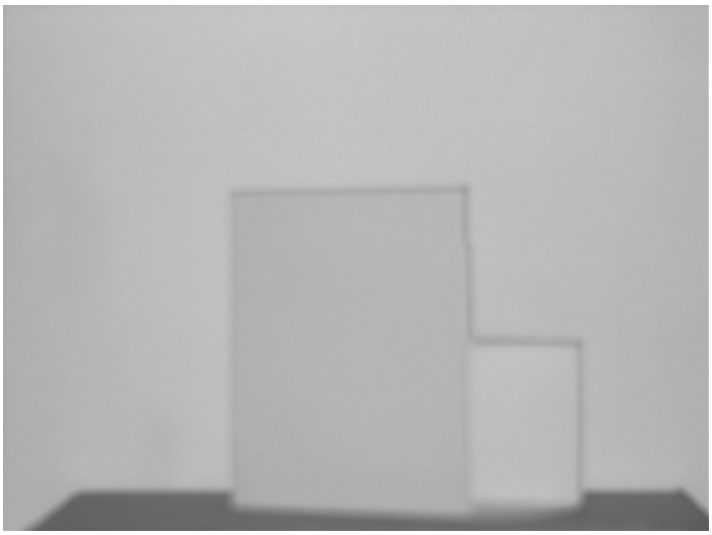


Figure 6.9 Image of sloping plane object

The mesh plot of the estimated depth map produced by the DfD system can be seen in Figure 6.10.

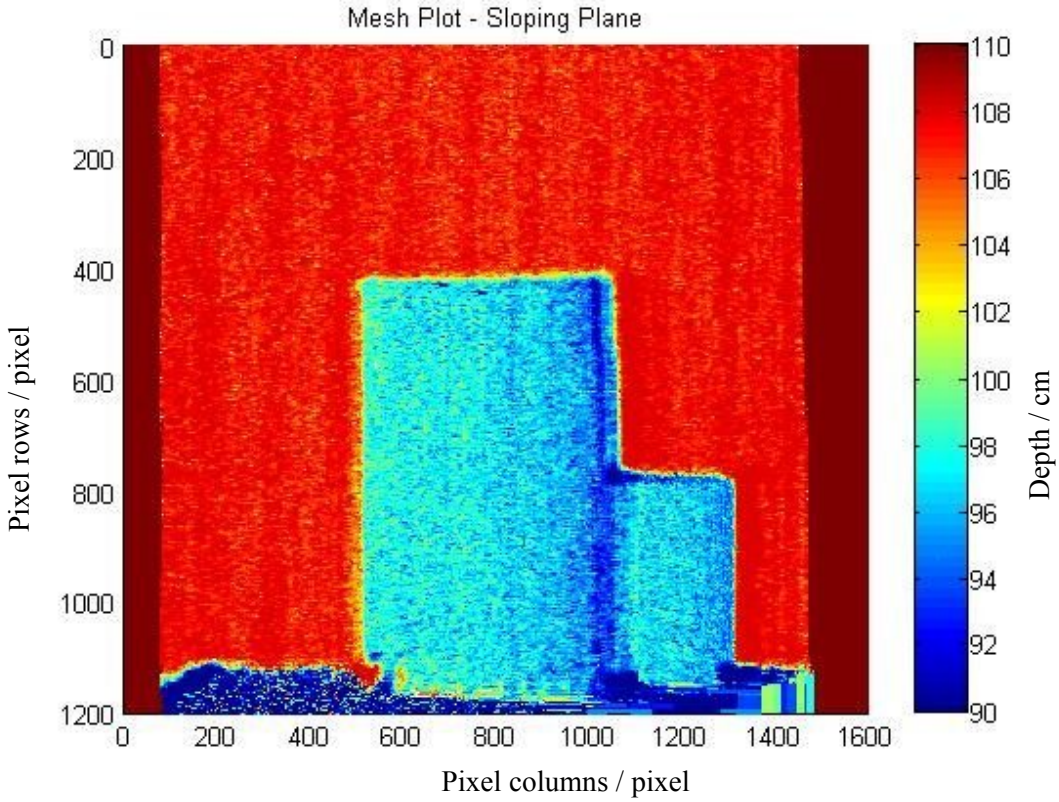


Figure 6.10 Mesh plot of sloping plane object depth map

Figure 6.10 shows that – despite some unwanted artefacts towards the bottom of the plot – the shape and varying depth of the sloping plane have been reasonably well

estimated. Even the interior of the open-end of the support box at the back of the panel shows approximately the correct degree of slope, being a slightly darker blue to the right of the box and progressing to a lighter blue towards the left-hand side.

The slope – being 45^{o} – is quite a reasonable test of the systems ability to continue to produce a reasonable depth estimate even when the angle of the object surface is not perpendicular to the optical axis.

Figure 6.11 shows the plot of a row sample across the width of the sloping plane, and a plot of the actual depth is included.

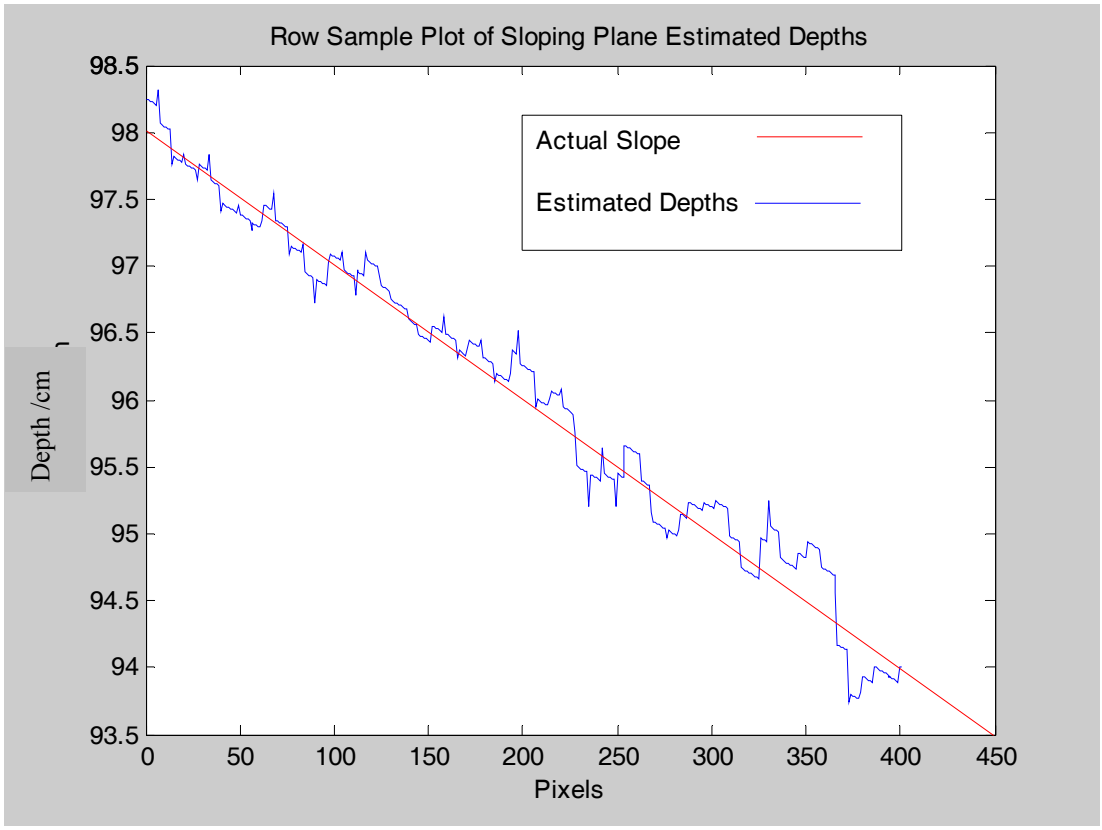


Figure 6.11 Plot of Sloping Plane Depth Estimates

A plot of the actual depth estimate error across the slope width is shown in Figure 6.12 below.

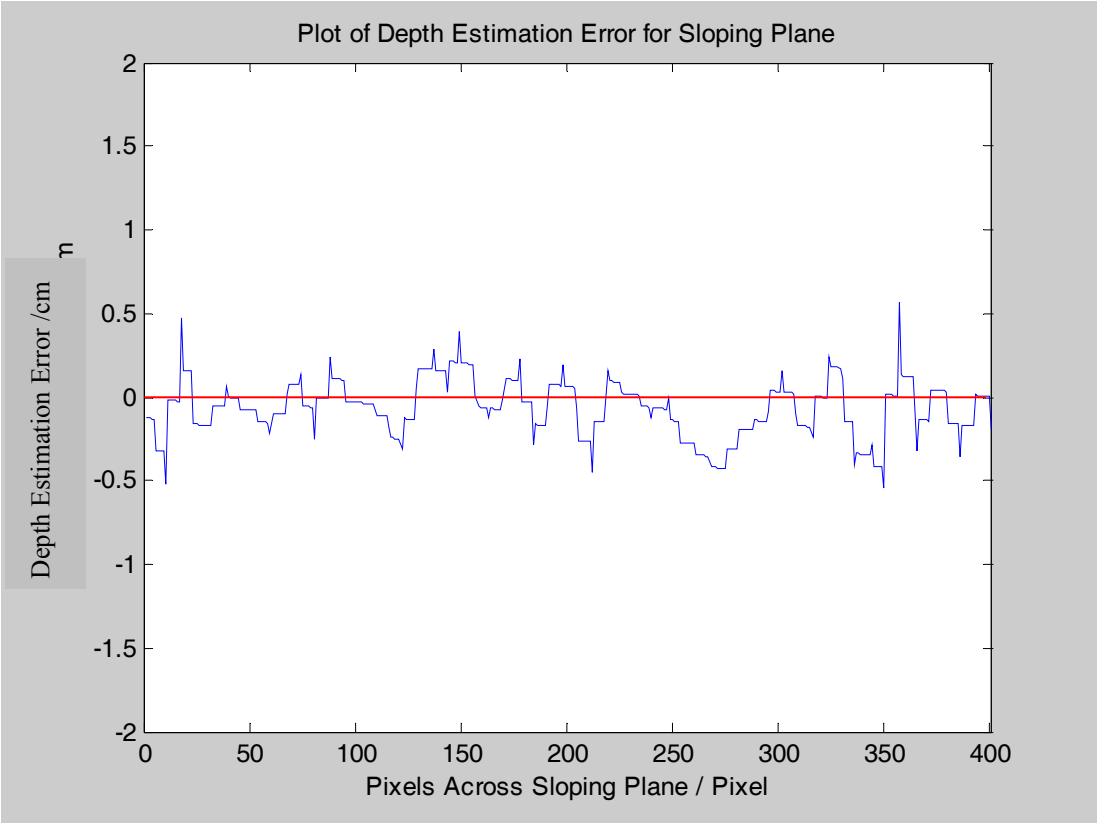


Figure 6.12 Depth Estimation Error for Sloping Plane Application

From the values used to produce this plot, a maximum error of approximately ± 0.53 cm was obtained with a variance of 0.0341. These results represent a slightly poorer level of accuracy than that obtained from the Varied Planes application, but given that the 45^{o} slope was quite a severe test of the system's ability to cope with non perpendicular surfaces they are quite acceptable.

6.2.3 Sphere of white polystyrene

This object offered a more severe test for a number of reasons,

a. The surface of the sphere was a rougher texture than the smooth material used in the previous two applications

b. The curvature of the sphere would exceed that tested in the previous two applications

c. There appeared to be the possibility that the surface of the sphere was more reflective than materials previously used, and that this could cause a degree of specular reflection in some areas of the sphere.

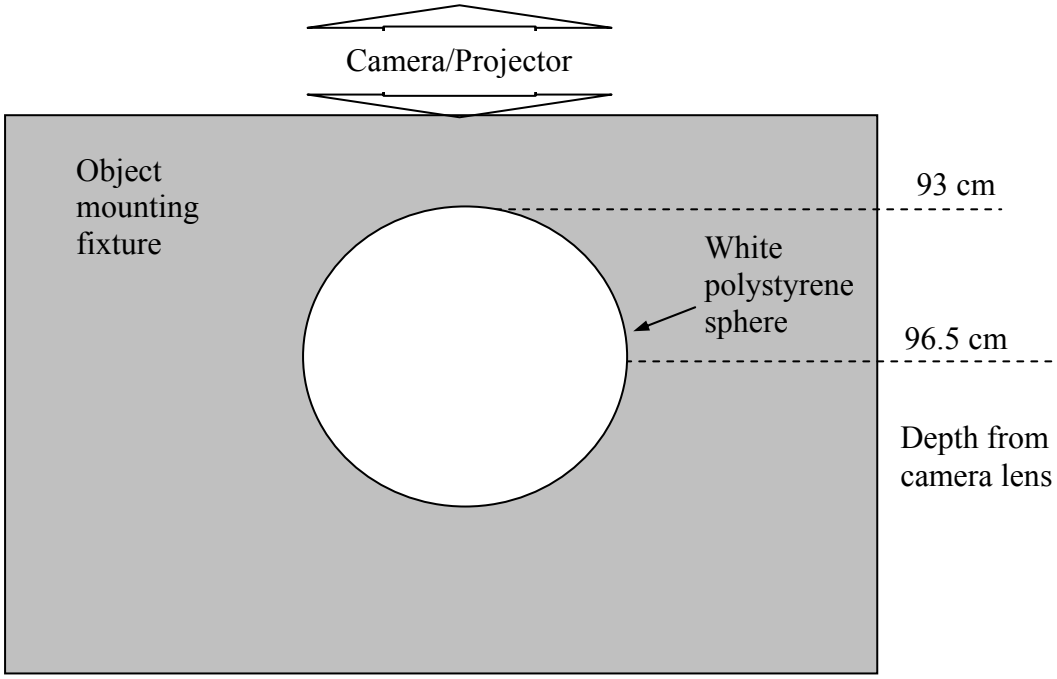


Figure 6.13 Plan view of spherical white ball shape

The sphere had a diameter of 7 cm, so it can be seen in Figure 6.13 that the maximum surface range visible to the camera/projector system would provide a depth range of 3.5 cm. However, section 5.4 of chapter 5 showed that the system only retains reasonable accuracy with a surface angle up to approximately 60^{o} from the normal to the optical axis.

Figure 6.14 shows the image of the sphere under non-active illumination.

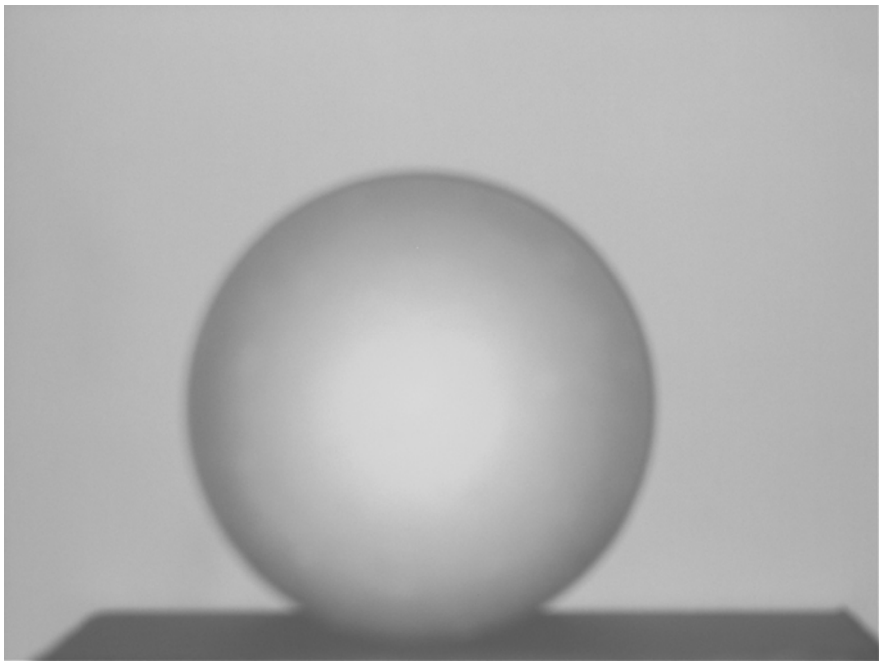


Figure 6.14 Image of white sphere object

The 4-D look-up table algorithm was applied to the white sphere, and the mesh plot of the estimated depth map is shown in Figure 6.15.

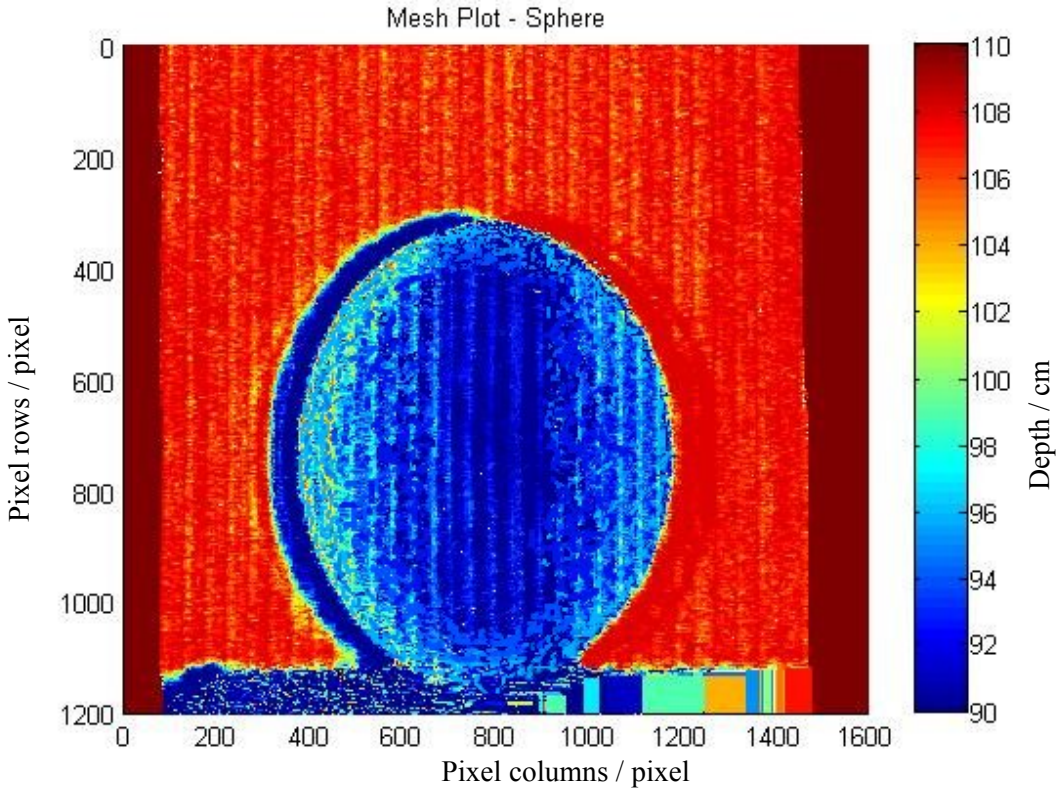


Figure 6.15 Mesh plot of white sphere object depth map

Figure 6.15 begins to show the expected limitations of the system when presented with a more complex shape. In one respect – considering the main central shape displayed – the system has coped reasonably well with the sphere in the sense that it has produced a clearly defined spherical shape, albeit one that appears smaller in diameter than the original. This is to be expected, as the pattern edge profiles become invalid around the periphery of the sphere once the surface angle is more than approximately 60^{o}. Given this fact, it is remarkable that the edge of the spherical shape reproduced is still so well defined.

Also, the gradient of the depth estimates within the spherical shape is also quite consistent with the expected depth profile defined in Figure 6.13. A degree of vertical striation is still evident in the depth estimate pattern, particularly at the shorter depths, but the estimates do tend to increase radially to a colour mapped depth of approximately 95 cm at the periphery. This would concur with the approximate depth of the spherical surface at an angle of 60^{o}, i.e. the natural limit of the systems measurable surface rotational range.

As a means of producing some quantitative estimate of the error, a sample of the depth map was extracted from a row that was positioned on the centre of the sphere. The sample length represents approximately 120^{o} of the sphere's surface (i.e. 60^{o} either side of the optical axis) as illustrated in Figure 6.16.

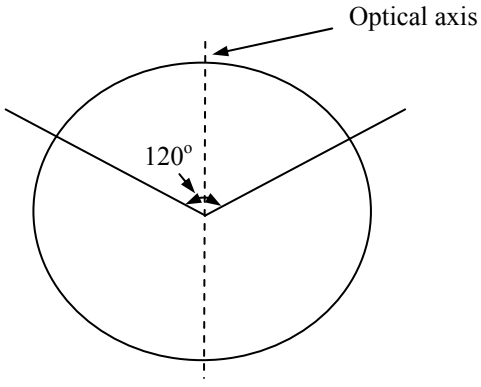


Figure 6.16 Sample range of extracted depth map row

From this row sample, the depth estimate plots shown in Figure 6.17 were produced.

Figure 6.17 Actual depth versus estimated depth for Sphere application

Figure 6.18 provides a plot of depth error derived by comparing the extracted depth estimate values with the values expected for each row pixel.

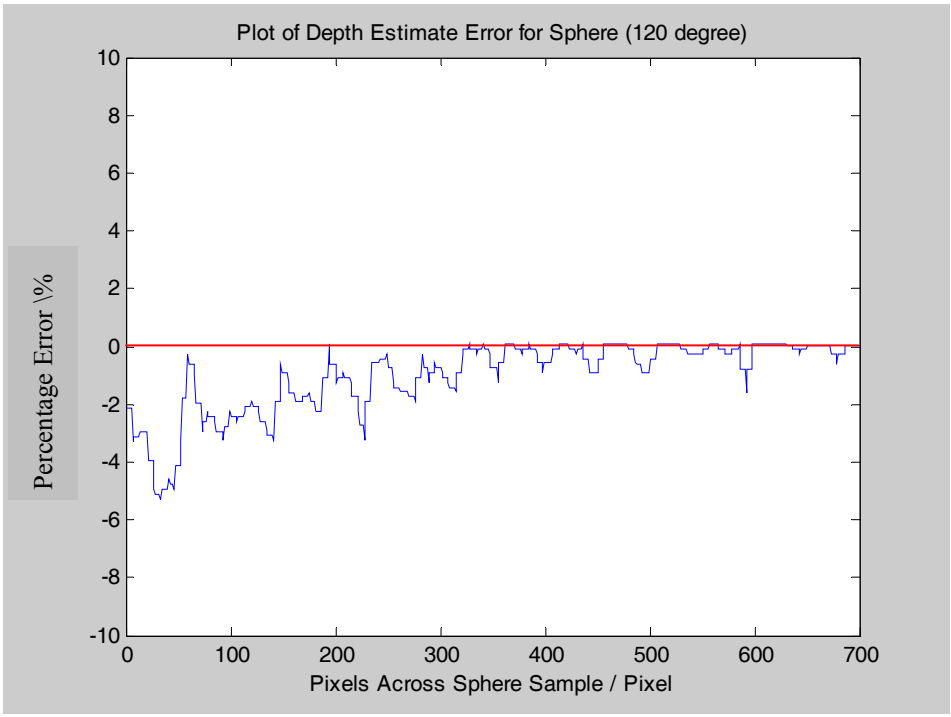


Figure 6.18 Depth Estimate Error for Sphere Application Sample

Both Figures 6.17 and 6.18 show the error on the left-hand side of the profile to be far higher than that on the right-hand side – which appears to retain a good level of accuracy (within 2%) through to the 60^{o} point of rotation on the right-hand side.

It is notable that this effect may be related to the existence of the curved artefact to the left of the sphere in the mesh plot (Figure 6.15). The reason for this unwanted artefact appearing in the depth map is not obvious, though it mirrors the shape of the periphery of the sphere so closely that it gives the appearance of a shadow or reflection. It may simply be the case that the backplane on the object mounting – only being a few centimetres behind the object – has produced a shadowing or secondary illumination effect. In which case, it would be trivial to correct the problem. The overall conclusion is that the system has performed well within its operating limits, but that it does not have the capability to reproduce the shape of objects that have curvature or surface angles greater than 60^{o} from the optical axis.

6.2.4 Complex shape – high-density (blue) polystyrene

This object was made on a rapid-prototyping machine and used various forms of sinusoidal machining patterns – in some cases these sinusoidal patterns were rotated 90^{o} from a previous pattern – so the end result was a complex 3-D profile of curved peaks and hollows[6.1]. An indication of the shape complexity can be seen in the photograph of Figure 6.19.

[6.1] Object supplied by Mr P Kimber (see Acknowledgments). Unfortunately the original machining pattern was not available for comparison.

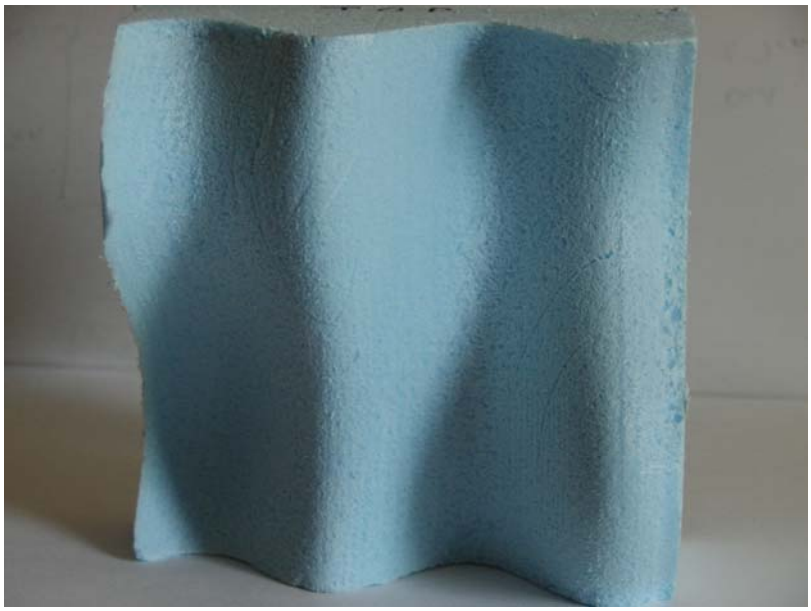


Figure 6.19 Complex shape – high-density polystyrene

The maximum variance in depth between the peaks and hollows of the shape was approximately 5 cm.

This shape represented quite a difficult challenge to the prototype system at its present stage of development. The initial image of the object taken by the DfD system using non-active illumination can be seen in Figure 6.20, and Figure 6.21 shows a mesh plot of the map of estimated depth values.

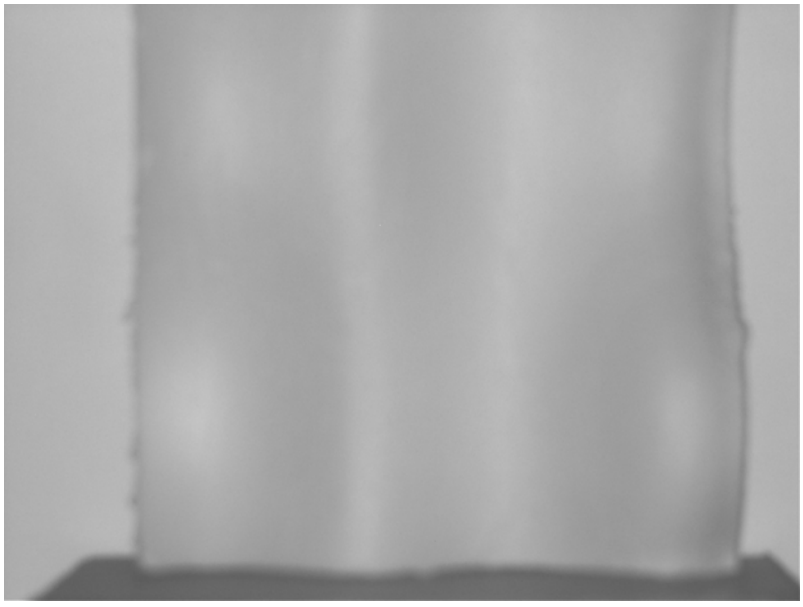


Figure 6.20 Image of complex shape under non-active illumination

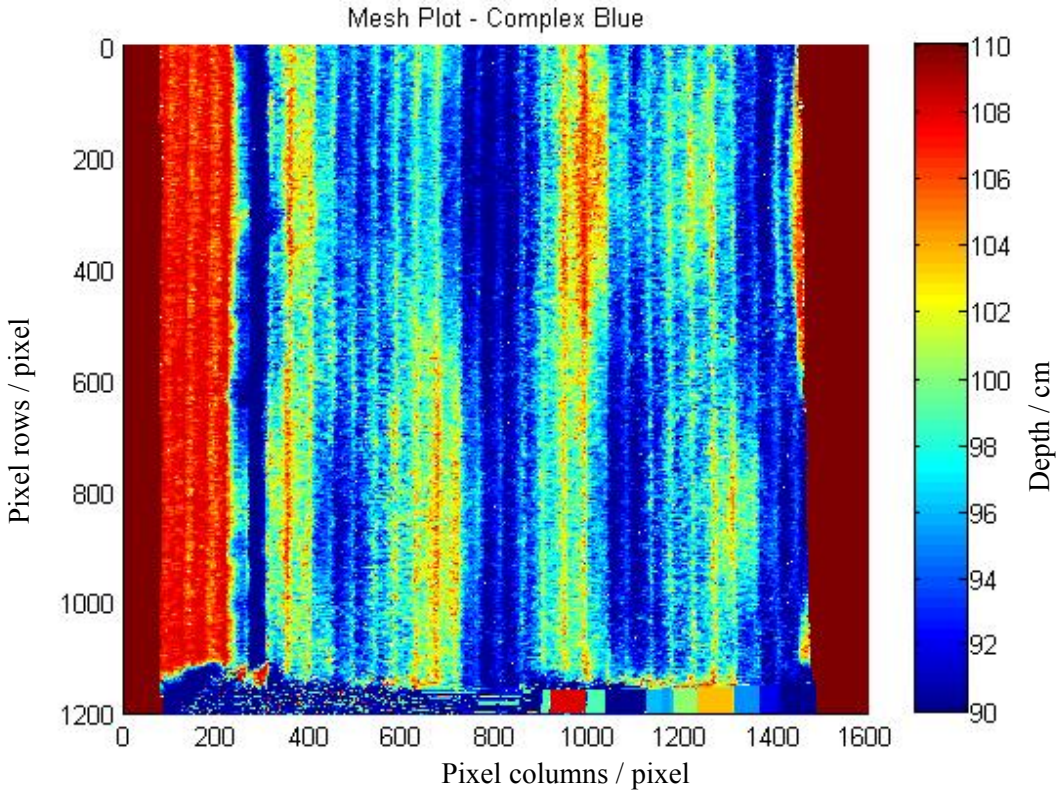


Figure 6.21 Mesh plot of depth map estimates for complex shape

A comparison of the mesh plot of Figure 6.21 with the photograph of Figure 6.19 shows that the system has actually made quite a good attempt at mapping the complex contours of the object. The general pattern of peaks and hollows appears to be relatively accurate when compared to the original object. However, it should be noted that the range of depths between minimum and maximum is shown in places to be somewhat greater than it should be. The colour mapping indicates peaks in the region of 92 cm to 94 cm range, whereas in parts of the hollow regions the colour map is entering the 'orange/red' range, i.e. 104 cm to 106 cm. In a limited number of areas, this would suggest a depth range closer to 12 cm rather than 5 cm.

It may be considered that this is the result of rather sharp changes in contour at the bottom of hollows (and at the top of peaks), and this may produce distortion of the

projected pattern edge profile causing an error in the depth estimate made at that point. It may also be the case that a degree of specular-type reflection is caused by these areas of the surface, and that this also tends to distort the projected pattern edge profile. Figure 6.20 does show the object exhibiting reflective areas at these points.

6.3 Summary of Chapter 6

The DfD system design was modified to provide projection of the structured light pattern along the optical axis. This ensured that the pattern illuminated the object on all surfaces areas viewed by the camera. The system was then re-calibrated in order to produce a new 4-D look-up table appropriate for this arrangement. Finally the system was applied to four 3-D objects of increasing surface contour complexity. Despite experiencing some problems that future developments of the system would need to compensate for, the system generated depth maps that were representative of the shape of the objects to a reasonable level of accuracy.

Chapter 7

Conclusions and Further Work

7.1 Conclusions

The principal aim of this research was to study the subject of depth from defocus (DfD), and to determine a particular DfD method that would be appropriate for the application of generating 3-D depth maps from 2-D images of objects.

The background motivation for this study was the possibility of producing a prototype DfD system that may be suitable for further development into a proprietary system for the non-contact measurement of manufactured artefacts, although it should be emphasised that the research was not specifically constrained to achieve this objective per se.

The specific work of the author has been to investigate the use of active illumination in a DfD system. The benefit of using active illumination was that the projected pattern dominates the scene in terms of variation of light intensity of the pattern features, and this can negate problems caused by viewing the object under passive light conditions. A relatively novel – yet computationally simple – means of using the shape of the defocused edge of the projected pattern was devised. This feature – or depth cue - varied relative to depth in a monotonic fashion, and gave a near linear response over the distance tested. It was also robust in the sense of presenting good resolution between depth intervals, so that the programming algorithm could reliably resolve the edge shape and relate it to a depth estimate.

This selected depth cue was used in the development of a high-resolution, spatially invariant, depth from defocus system that incorporated a number of novel features.

A 2-D look-up table system was tested over the range of 90 cm to 114 cm inclusive (i.e. the 2-D look-up table had 25 elements representing depths in this range at 1 cm intervals). The results were presented in Table 5.1 (and the associated plot in Figure 5.4) on page 5-5. The error in depth estimate ranged between 0.382% at 90 cm and 1.724% at 102 cm. This gave an average error over the range of 1.282%, which compared very favourably with other research reviewed in chapter 3. Of the small number that have used a single-image technique, only Cardillo [1] claimed an error of 1.9%, but that resulted in a sparse depth map as it used sharp edge features in the image.

When investigating the effect of *variation in the property of object surface reflectance* (or light absorption) on the level of error in depth estimation, it was found that with surface grey levels from 0% up to 50% the depth estimate error remained below approximately 2%. For surfaces darker than 50% grey level, the error increased significantly – to over 5% at about 95% grey level. This determined that development of the system should continue to use objects of white or relatively light hue, and the problem of applying the system to dark objects would need to be addressed during further work.

A second factor to be investigated was that of the degree of *rotation of the object surface*. Figure 5.8 showed that the system displayed a high degree of tolerance to surface rotation, with very little increase in depth estimate error until the surface was rotated approximately 60° from the perpendicular. Beyond this point, the error increased rapidly, and was approaching 6% when the surface was rotated by an angle of 75°. Whilst this demonstrated a very positive result for the performance of the system, it must still be recognised that some complex shapes with curves at angles of 60° or greater will inevitably not have those parts of their surface area represented in

the eventual depth map produced by the system. However, this limitation would apply to almost any form of optical metrology system, unless some technique (e.g. rotating the object) was incorporated into the system.

An investigation into the possible benefits of *image averaging* was also carried out to determine whether this would improve depth estimate accuracy. Experiment 7 in section 5.5 of chapter 5 captured 16 time-lapsed images of the same image scene of the projected pattern at a fixed depth. Figure 5.9 showed the results of averaging each successive image. These results showed no marked improvement of the depth estimate as the number of averaged images increased. The original image produced a depth estimate with an error of approximately 0.47%, and – apart from one outlying reading with 3 averaged images – the estimated error continued to fluctuate around this value. It was assumed that the reason image averaging did not generate any improvement was in this case simply due to the fact that the projected structured light pattern was so dominant over other image features – including noise – that removal of noise effects by this method has little consequence in any case.

One element that was suspected of being a possible source of variability in the system was the 'Plus U2' data projector. This device was vital in enabling one of the most novel features of the system eventually devised by the author – that of moving the projected pattern during object analysis. However, whilst this was a relatively high-specification device, it was still not designed for this specific type of use and section 5.6 of Chapter 5 explained where some of these features could cause error due to variability in the operation of the system. Never the less, the author was required to use the equipment that was available, and the functionality of the device still enabled the concept system to be developed and tested satisfactorily.

The *spatial invariance* of the system was improved by a number of design features. The number of projected pattern edges per image row was maximised by optimization of the pattern size. This was empirically determined to be ten edges per row across the width of the image.

The novel technique of *moving the pattern* and taking a succession of images was developed. As twenty movements of the pattern could be achieved within one period of the pattern's design, this meant that the number of edges – and hence number of depth estimates – per row was increased from ten to two hundred. Section 5.7 of chapter 5 described the need for this development, and section 5.8 described how it was achieved. In practice, it required a relatively complex programming algorithm to arrange the 200 edges (from 20 pattern-shifted images) in the correct positional order. Section 5.9 and Figure 5.22 described the very significant improvement in consistency of depth estimation across the image width, where a variation of less than ± 1% is calculated. A 3-dimensional look-up table was now required to map these depth estimates (i.e. profile vector x image width position x depth estimate). The 3-D look-up table held 17 x 200 = 3400 standard profiles. However, when analysing an image of unknown depths, the 'column' position of each extracted edge profile was known, hence the search algorithm was only required to compare the extracted profile with 17 profiles in each depth element that represented that particular row x column position.

In order to improve the spatial invariance of depth estimates between image rows (or along the length of the image) the 4-D look-up table system was devised to hold 17 x 200 x 200 = 680,000 standard profiles. The general structure of the 4-D look-up table is illustrated in Figure 5.23.

It should be noted that, although the 4-D look-up table had vertical resolution in terms of 6 rows per standard profile, when an image of an object was being analysed each row was analysed individually. This simply meant that, e.g. rows 1 to 6 used the standard profile in vertical block 1, etc. The horizontal position of each extracted profile along the row determined which 'column' position of the table was used for the search.

This meant that, using the 4-D look-up table system, a total of 200 x 1200 = 240,000 depth estimates were made over the entire image area, and each estimate was constrained to only search through 17 standard profiles that have been devised for the specific 6 x 8 pixel window in which the extracted edge profile was located. This made the system spatially invariant and effectively offered a solution to one of the remaining problems that has existed in depth from defocus systems to date.

Testing the system over a depth range of 10 x 1 cm intervals (90 cm to 99 cm), the results in Appendix B-11, illustrated in the plot of Figure 5.26, showed an average percentage RMS error of approximately 1.5%, and a depth estimate error of well below ±0.5%. This compared very well with, for example Ghita and Whelan [2] who claimed 3.4% error over a distance of 86 cm. It should be noted, however, that this was achieved with a two-image system.

The use of *texture neutralisation* (see section 5.12 *Neutralising image texture to improve depth estimate accuracy*) was devised to negate the effects of reflections and other possible unwanted artefacts (e.g. from an imperfect data generator).

Figure 5.27 showed a significant improvement in both the percentage depth estimation error, and the standard deviation of estimates per depth across the 10 cm range used by the 4-D look-up system. The average percentage error over the range was calculated to be 0.0356%, with standard deviation of 0.0894 for the same

percentage error values. These results showed significant improvement over other previous research – including those using two-image techniques. Pentland, et al [3] claimed 0.5% RMS error using a two-image look-up system, but with low spatial resolution in terms of depth estimates per image. Nayar, et al [4] claimed 0.24% using a two-image system, but with a spatial resolution of 256 x 240 = 61,440 depth estimates per image. There are no single image systems that make any comparable claims in terms of accuracy or spatial resolution based on direct measurement evidence.

Section 7.2 (below) summarises the main features that are considered to be novel and that provide original contribution to the field of depth from defocus.

7.2 Summary of novel features and original contribution

These include,

1. The concept of moving the projected pattern during object analysis in order to increase the spatial resolution of depth estimates.
2. The concept of a 4-dimensional look-up table that presents a set of standard edge profiles specific to each 6 x 8 pixel window, and hence provides a very high degree of spatial invariance.
3. A texture neutralisation algorithm that negates the effect of texture features in the original image.
4. A system that, due to the calibration scheme, gives results that are completely independent of the point spread function of the camera lens arrangement, and of any other blurring or sampling effect of the image acquisition system.

5. A system that does not inherently depend on high levels of accuracy in the set up and alignment of the original projector/camera arrangement – provided the same set up is retained between calibration and application.
6. An analysis time of approximately 14 seconds to make 240,000 depth estimates (Pentium(R)4 CPU, 3.00GHz, 512 MB RAM).

7.3 Suggestions for further work

A number of areas of improvement and/or modification could be considered.

1. Removing the 'vertical striation' effect that seemed to occur at smaller depths (i.e near to the 90 cm depth level). The problem with this effect is that it appeared to occur at a period of about 1/30 of the image width, or approximately every 50 pixels along the image row. If the cause of the effect was related to the spacing of depth estimates across each row of the depth map, then this does not correlate very well as depth estimates are entered into the depth map at horizontal spacing of approximately 6 to 7 pixels. The fact that this effect appears to be more prevalent at depths nearer to 90 cm could indicate some form of interference pattern related to the light path of the original projected image and the reflected image rays travelling towards the camera lens. Further investigation of this could determine whether this was the case, and find ways of eradicating or avoiding the effect.
2. An improvement to the data projector as a means of providing a moveable projected pattern. This could eradicate many of the problems regarding this device, as discussed in Chapter 5. It may be possible to design some form or mechanism carrying an etched plate on a micro-adjustable movement. This could then be built into a customised form of projector – possibly illuminated

by e.g. a xenon lamp with adequate diffraction filtering – and the whole system could be automated such that pattern movement and capturing/saving the image would be entirely synchronised.

3. A minimised form of pre-calibration could be devised, whereby some known feature – effectively acting as a form of fiducial marker – would be seen by the camera in addition to the object under analysis. The system would estimate the depth of this artefact against its known depth and would generate a compensating value that could be used to 'trim' or 'fine tune' depth estimates made on the object in view. A similar mechanism could also be considered as a means of compensating for variations in the colour temperature of the object surfaces.
4. More efficient search algorithms and/or methods of storing the profile data could be explored. As was stated in the introduction to this thesis, this was never intended to operate as a real-time system, but improvements could be made to the overall operating speed of the system. Currently, the processing of an image to produce a depth map takes approximately 14 seconds. Of course there is no reason why the full resolution of the system (240,000 depth estimates) should always have to be used. It would be a trivial modification to introduce a user-controlled variable that determined whether the system evaluated every one of the 1200 rows in the image or e.g. every 5^{th} row. Many applications may not require the full resolution available. It may be possible to carry out some pre-processing of the image in order to determine what features exist, and control the system such that it only extracts and analyses edge profiles from selected areas. This could make the system very

fast, although obviously the pre-processing time would need to be factored into the calculation.

5. Although the system does not suffer from so called 'windowing effects' (i.e. where a window under analysis for its gradient of defocus is affected by the defocus of neighbouring windows), there is a 'border effect' where pattern edges close to either the left or right hand borders of the image are ignored. This is due to the fact that they may provide only a partial profile if they are too close to the edge, and so are not considered valid. This is not a major problem if the user does not expect to process an object that it very close to the left or right hand side of the image, but it may be possible to devise a method of extrapolation of the valid depth estimates near the edges to effectively fill the gap in that small boundary area.
6. To cope with objects of more complex shapes (as demonstrated in chapter 6), it may be possible to devise a means of rotating the object, and capturing a series of images that includes the 'moving pattern' concept. This would constitute quite a significant development over the current system.
7. All program development and testing was carried out in MATLAB. It would be possible to improve the computational efficiency, and hence speed of processing if the program algorithms were re-written in a high-level language such as 'C'.

Chapter 7 – References

[1] Cardillo J, et al, "3-D Position Sensing Using a Single-Camera Approach", *Midwest Symposium on Circuits and Systems*, *pp325-328,* 1990

[2] Ghita O Whelan P F, "A Video-rate Range Sensor Based on Depth From Defocus", *Opt Laser Technol; v33, No3, pp167-176*, 2001

[3] Pentland A P, Scherock S, Darrell T, Girod B, "Simple range cameras based on focal error", *J.Opt.Soc.Am; Vol.11 No11; pp.2925-2934*, 1994

[4] Nayar SK, Watanabe M, Noguchi M, "Real-time Focus Range Sensor", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *Vol.18 No12, pp1186-1197,* 1996

Appendix A-1

Note: *This is a later version of the VerticalWhiteLines.m file. Earlier versions did not use the 'timed shift' facility produced by the 'while' loop in this version.*

```
%VerticalWhiteLines - An M-file to generate a black/white vertical stripe pattern
%Stripe size can be varied to suit required projected structured light pattern
%W E Crofts, Image Processing and Expert Systems Group, University of Warwick

clear
close 'all'

%Create rectangular grey cell (16 x 20 pixels)
C1(1:16,1:20) = [0.7];

%Create rectangular black cell (16 x 20 pixels)
C2(1:16,1:20) = [0];

%Adjoin C2 and C1 to make 16 x 40 black/white matrix
C4=[C2 C1];

%MATLAB function 'repmat' can now be used to tile this checkered cell into a larger
%grid
%i.e. A 480x640 pixel vertical stripe pattern
GridA=repmat(C4,30,16);
%NOTE: The X and Y multipliers in 'repmat' can be varied to produce a rectangular
%grid

imshow(GridA)
beep on
%return
set(gcf,'Position', [1 31 1024 666]);

z=1;
while z == 1
   imagesc(GridA,[0,1])
   set(gcf,'Position', [1 31 1024 666])
   drawnow
   t = timer;
   set(t, {'StartDelay', 'Period'}, {5, 1});
   t.TimerFcn = 'disp("Processing...")'
   start(t);
   wait(t);
   GridA = circshift(GridA,[0 5]);
   beep

end;
```

Appendix A-2

Note: ***This is a later version of the VertStripeCalc.m file. A number of iterations of the program were produced as the algorithm to investigate the image of the structured projected pattern developed.***

```
% VertStripeCalc_v2  -  An M-file to analyse the variation of stripe parameters at
%different depths for a projected pattern of vertical white stripes
%W E CROFTS, Image Processing & Expert Systems Group,University of Warwick


[infile,pathname]=uigetfile('\Research\PhD\MATLABfiles\structuredlight\060706U
EyeIntensityVar\*.bmp','Please select a file');
%UIGETFILE - Interactively retrieves a filename into 'infile' by displaying a dialog
%box.

infile1 = [pathname infile];
        %Associates full path and filename to INFILE1

[image1,map] = imread(infile1);
     %[X,MAP] = IMREAD(FILENAME,FMT) reads the indexed image in
     %FILENAME
   %into X and its associated colormap into MAP. Colormap values in the
   %image file are automatically rescaled into the range [0,1].

%disp('The image will now be converted from RGB to Intensity type')
%disp('Remove this command when not using a digital camera')
%image2=rgb2gray(image1);
%clear image1

[M,N] = size(image1);          %M = row, N = column

%Determine sample row to be scanned.
R = M/4;        %Determine row to be scanned
P = N/2;        %Determine centre of sampled row

profile1=(image1(R, 1:N));    % Creates sample from selected row

%Analyse row sample for projected stripe width
ave=((max(profile1) + min(profile1))/2);
%Finds average intensity value, i.e. half profile height

% Finds stripe nearest to centre of sample row
% and determines width and standard deviation.

if profile1(P) >= ave  %If centre position is WITHIN WHITE STRIPE
   %% Find global position of local minima to left of stripe %%
   while profile1(P) >= ave,   %Is current position still within stripe?
      P = P - 1;               %Move to left to find start of white stripe
   end
```

```
   Gright=P;                    % Note rightmost position of 'gap'
   while profile1(P) <= ave,    % Find Leftmost position of 'gap'
      P = P - 1;                % Move to left to find start of 'gap'
   end
   P = P+1;
   Gleft = P;                   % Note leftmost position of 'gap'
   [C,I]=min(profile1(Gleft:Gright));        % C=local minima, I = local position of
                                             % minima
   S_start = Gleft + I;         % Global position of minima - start of full stripe width

   %% Find global position of local minima to right of stripe %%
   P = N/2;                     % Re-centre cursor
   while profile1(P) >= ave,    %Is current position still within stripe?
      P = P + 1;                %Move to right to find start of 'gap'
   end
   G2left = P;                  % Note leftmost position of 'gap'
   while profile1(P) <= ave,    % Find Rightmost position of 'gap'
      P = P + 1;                % Move to right to find end of 'gap'
   end
   P = P-1;
   G2right = P;                 % Note rightmost position of 'gap'
   [C,I]=min(profile1(G2left:G2right));      % C=local minima, I = local position of
                                             % minima
   S_end = G2left + I;          % Global position of minima - end of full stripe width

   % Full width of single stripe being analysed now commences at
   % 'profile1(S_start)' and ends at 'profile1(S_end)'
   profile2 = profile1(S_start:S_end);
   %plot(profile2)

else            %If centre position is WITHIN 'GAP'
   %% Find global position of local minima within 'gap' %%
   while profile1(P) <= ave,            % Find Rightmost position of 'gap'
      P = P + 1;                        % Move to right to find start of white stripe
   end
   P = P-1;
   Gright = P;                          % Note rightmost position of 'gap'

   while profile1(P) <= ave,            % Find Leftmost position of 'gap'
      P = P - 1;                        % Move to left to find start of 'gap'
   end
   P = P+1;
   Gleft = P;                           % Note leftmost position of 'gap'
   [C,I]=min(profile1(Gleft:Gright)); % C=local minima, I = local position of
                                        % minima
   S_start = Gleft + I;         % Global position of minima - start of full stripe width

   %% Find global position of local minima to right of stripe %%
   P = Gright + 1;              % Place cursor back to start of white stripe
   while profile1(P) >= ave,    %Is current position still within stripe?
```

```
    P = P + 1;                  %Move to right to find start of 'gap'
  end
  G2left = P;                   % Note leftmost position of 'gap'
  while profile1(P) <= ave,     % Find Rightmost position of 'gap'
    P = P + 1;                  % Move to right to find end of 'gap'
  end
  P = P-1;
  G2right = P;                  % Note rightmost position of 'gap'
  [C,I]=min(profile1(G2left:G2right));      % C=local minima, I = local position of
                                            % minima
  S_end = G2left + I;           % Global position of minima - end of full stripe width

  % Full width of single stripe being analysed now commences at
  % 'profile1(S_start)' and ends at 'profile1(S_end)'
  profile2 = profile1(S_start:S_end);
  %plot(profile2)

end

  %% OUTPUTS %%

  % For 'relative pulse height'
  h = (max(profile1) - min(profile1));
  H = num2str(h);
  disp(['Pulse height relative to minimum level = ',H]);

  % For 'standard deviation' of full stripe profile
  s = std(profile2);
  S = num2str(s);
  disp(['Standard Deviation of total stripe profile = ',S]);

  % For period of pattern frequency (stripe+gap width) at average height
  % intensity
  period = G2left - Gleft;
  Period = num2str(period);
  disp(['Period of pattern frequency in pixels = ', Period])

  % For 'Stripe width' using 15% of height intensity.
  Wlev = ((max(profile2) - min(profile2)) * 0.15) + min(profile2);
  Le = length(profile2);
  figure(1)
  hold on;
  plot(1:Le,Wlev,'c+');         % Plot 15% level onto previously 'held' profile2 plot
  hold off;
  count=0;
  for i=1:Le,
    if profile2(i)>=Wlev
      count=count+1;
    end
  end
```

```
%Mark Space Ratio - (pixels > 15% threshold / pixels <threshold)
msr = (count / (Le - count));
MSR = num2str(msr);
disp(['Mark-Space Ratio at 15% of height intensity = ', MSR]);
disp(' ')
disp(' ')


% For profile of stripe edge - i.e. defocus effect
P = Gright + 1;                 % Point of ave intensity on 'rising edge' of stripe
while profile1(P) > profile1(P-1),  %Is edge curve decreasing?
   P = P-1;                         %Move cursor left
end
StartCurve = P;
P = Gright + 1;                     %Reset position of P
while profile1(P) < profile1(P+1), %Is edge curve increasing?
   P = P+1;                         %Move cursor right
end
EndCurve = P;
profile3 = double(profile1(StartCurve:EndCurve));
CurveData = double([1:1:length(profile3)]);          %Req'd for curve fitting
                                                     % figure
plot(profile3, 'ko');

disp(['Number of samples in curve profile = ', num2str(length(profile3))]);
ave
close all

%clear all;

fclose ('all');
        %FCLOSE  -  Close a file.
        % FCLOSE('all') closes all open files, except 0, 1 and 2.
```

Appendix A-3

Note: ***This is a later version of the AverageCalc.m file. It also demonstrates features such as edge detection and extraction, and smoothing of the row profile to aid this process.***

```
% AverageCalc.m  -  An M-file to extract a number of edge profiles at fixed
% depth, and average them to produce a 'best estimate' profile for that depth.
% Edge profiles for specific depths are generated using a vertical stripe
% projected light pattern and stored in look-up table LUT.

% W. E. CROFTS, Image Processing & Expert Systems Group, University of
% Warwick
%
clear

Gate = input('Load another image? Enter 1 to continue or 0 to end >');
Im = 1;                 % Set image counter
while Gate == 1         % User control for number of image profiles to be stored
                        % in look-up table

   %%LOAD IMAGE VIA GUI WINDOW%%
   [infile, pathname]=uigetfile('\Research\PhD\MATLABfiles\structuredlight\
    *.*','Please select a file');
        %UIGETFILE - Interactively retrieves a filename into 'infile' by displaying a
        %dialog box.

   infile1 = [pathname infile];
        %Associates full path and filename to INFILE1

   [image1,map] = imread(infile1);
        %[X,MAP] = IMREAD(FILENAME,FMT) reads the indexed image in
        %FILENAME into X and its associated colormap into MAP. Colormap
        %values in the image file are automatically rescaled into the range [0,1].

   profile = [];        %Create empty vector for final averaged profile
   count = 1;           %Initialise count for no of profiles
   %See 'averager' function for use of above 2 commands


   %% RECORD ALL EDGE CURVE SAMPLES ON EACH ROW %%

   [M,N] = size(image1);            %M = row, N = column
   for m = 600 %:20:(M-20)          % Sample Rows in image
      profile1=(image1(m, 1:N));    % Creates sample from selected row
      profile1=smooth(double(profile1)); %Smooth 'profile1' to eradicate 'glitches'
      ave=((max(profile1) + min(profile1))/2);
      %Finds average intensity value, i.e. half profile height
```

```
%Identifies first 'rising' edge in row
n = 1; % Column counter to start of row
if profile1(n) >= ave      %If first pixel is WITHIN WHITE STRIPE
   % Find position of first FULL 'rising' edge to right of black stripe
   while (profile1(n) >= ave) & (n < N)          %Is current position still within
                                                 %white stripe?
      n = n + 1;             %Move to right to find end of white stripe
   end
   while (profile1(n) < ave) & (n < N)           %Is current position still within
                                                 %black stripe?
      n = n + 1;             %Move to right to find end of black stripe
   end
   U1 = n;                   % Note position of first full rising edge

elseif profile1(n) < ave   %If first pixel is WITHIN BLACK STRIPE
   % Find position of first FULL 'rising' edge to right of black stripe
   while (profile1(n) < ave) & (n < N)           %Is current position still within
                                                 %black stripe?
      n = n + 1;             %Move to right to find end of black stripe
   end
   while (profile1(n) >= ave) & (n < N)          %Is current position still within
                                                 %white stripe?
      n = n + 1;             %Move to right to find end of white stripe
   end
   while (profile1(n) < ave) & (n < N)           %Is current position still within
                                                 %black stripe?
      n = n + 1;
   end
   U1 = n;                   %Note position of first full rising edge
end

%Identifies last 'rising' edge in row
n = N;                        %Column counter to end of row
if profile1(n) >= ave      %If last pixel is WITHIN WHITE STRIPE
   % Find position of last FULL 'falling' edge to left of white stripe
   while (profile1(n) >= ave) & (n > 1)          %Is current position still within
                                                 %white stripe?
      n = n - 1;             %Move to left to find start of white stripe
   end
   while (profile1(n) < ave) & (n > 1)           %Is current position still within
                                                 %black stripe?
      n = n - 1;             %Move to left to find start of black stripe
   end
   while (profile1(n) >= ave) & (n > 1)
   n = n - 1;
   end
   UL = n;                   % Note position of last full rising edge
elseif profile1(n) < ave   %If last pixel is WITHIN BLACK STRIPE
   % Find position of last FULL 'rising' edge to left of black stripe
```

```
        while (profile1(n) < ave) & (n > 1)          %Is current position still within
                                                     %black stripe?
           n = n - 1;            %Move to left to find start of black stripe
        end
        while (profile1(n) >= ave) & (n > 1)         %Is current position still within
                                                     %white stripe?
           n = n - 1;            %Move to left to find start of white stripe
        end
        UL = n;                  %Note position of last full rising edge

     end

     edgepos = U1;               %Start at first full rising edge
     while edgepos <= UL
        [profile2] = RisingCurve(edgepos,ave,N,profile1);
        [profile, count] = averager(profile2, profile, count);
        NextEdge = Riseseek(edgepos,ave,profile1,N);
        edgepos = NextEdge;
     end
  end

  %Save sum of averaged profile in look-up table 'LUT'
  LUT(Im,:) = profile;
  disp(' ')
  Im2 = num2str(Im);
  disp(['Image ' infile ' has been stored at location ' Im2 ' in LUT'])
  disp(' ')
  Gate = input('Load another image? Enter 1 to continue or 0 to end >');
  Im = Im + 1; %Increment image counter (for shifted pattern projections)
end

save('LookUpTable3','LUT')


%*****************************************************************%




fclose ('all');
        %FCLOSE  -  Close a file.
        % FCLOSE('all') closes all open files, except 0, 1 and 2.
```

Appendix A-4

RisingCurve.m – function to determine the boundaries of the edge profile, extract the edge profile from the row profile, and normalise it in 'y'

```
function [profile2] = RisingCurve(edgepos,ave,N,profile1)
   % For profile of stripe edge on rising edge - i.e. defocus effect
   P = edgepos - 1;              %Move cursor back into black stripe

   %Find centre of 'low' and 'high' peak either side of rising edge
   while (profile1(P) < ave) & (P > 1)        %Is current position still within black
                                              %stripe
      P = P - 1;                 %Move to left to find start of black stripe
   end
   Fall1 = P;                    %Mark start of black stripe
   StartCurve = round((edgepos + Fall1) / 2);        %Profile starts at low peak

   P = edgepos;                  %Reset cursor 'P'
   while (profile1(P) >= ave) & (P < N)
      P = P + 1;                 %Move to right to end of white stripe
   end
   Fall2 = P;                    %Mark end of white stripe
   EndCurve = round((edgepos + Fall2) / 2);          %Profile ends at high peak

   profile1a = profile1(StartCurve:EndCurve);        %Interim profile vector

   %% NORMALISE CURVE IN 'Y' BETWEEN 0 AND 1 %%
   Min = min(profile1a);
   Max = max(profile1a);
   L = length(profile1a);
   %ERROR CHECK
   Denom = Max-Min;
   if Denom == 0
      disp('DIVIDE BY ZERO ERROR DETECTED')
      disp('EXAMINE WORKSPACE VARIABLES BEFORE CONTINUING')
      disp('Program will continue on any key press')
      pause
   end

   for x = 1:L
      profile1b(x) = (profile1a(x) - Min) / (Max - Min);
   end

   profile2=profile1b;
```

Appendix A-5

averager.m – function to normalise the extracted profile in 'x', and to add it to an 'accumulator' profile, hence producing an average for the set.

```
function [profile, count] = averager(profile2, profile, count)
% Averager.m  -  Increases 'profile2' vector length to '140' by 'padding'
% with 1's at 'EndCurve' and 0's at 'StartCurve'. Vector is then shifted
% right to fix 'EndCurve' position to right-hand end of vector.
% The 40 values on the 'left-hand' of the vector are then removed to 'trim'
% the profile to a length of 100 values.
% This process allows the profile form to carry information on both the
% shape of the edge profile itself and on the width of the white bar in the
% projected pattern.  The latter being proportional to depth as the white
% bar defocuses as the projected image moves nearer than to the camera lens
% All subsequent profile vectors are then added to a 'profile' accumulator
% to produce an averaged profile for the presribed image area at that depth


% W. E. CROFTS, Image Processing & Expert Systems Group, University of
Warwick

X1 = length(profile2);
if rem ((140-X1),2) == 0                %If 'even' no of elements in profile
   Subzeros = zeros(1,(140-X1)/2);   %Set vector of 'zeros' for shift mask
   Subones = ones(1, (140-X1)/2);    %Set vector of 'ones' for shift mask
else    % If 'odd' no of elements in profile
   Subzeros = zeros(1,(140-X1+1)/2);          %Set vector of 'zeros' for shift mask
   Subones = ones(1, ((140-X1+1)/2)-1);       %Set vector of 'ones' for shift mask
end
NewXframe = [Subzeros profile2 Subones];           % Increase vector length to 140

%Shift profile to locate 'EndCurve' at right-hand end of vector
Shift = length(Subones); %Determine required 'right' shift
profile3 = circshift(NewXframe,[0,Shift]);    %Shift 'EndCurve' to rightmost end
profile3 = profile3(41:140);    %Remove extraneous elements from 'left' of vector

if count ==1
   profile = profile3;
   count = count + 1;
elseif count > 1
   profile = (profile + profile3) / 2;
   count = count + 1;
end
```

Appendix A-6

Riseseek.m – function to find the position of the next valid 'rising' edge in the image row.

```
function NextEdge = Riseseek(edgepos,ave,profile1,N)
%Find next rising edge on profile1
%Also return right edge of current white stripe e.g. for width check
while (profile1(edgepos)) >= ave & (edgepos < N)   %Is current position still
                                                    %within white stripe?
   edgepos = edgepos + 1;              %Move to right to find end of white stripe
end
Rightedge = edgepos;           % Temporary command to return width of white stripe
while (profile1(edgepos) < ave) & (edgepos < N)     %Is current position still
                                                     %within black stripe
   edgepos = edgepos + 1;              %Move to right to find end of black stripe
end
NextEdge = edgepos;                    % Note position of next rising edge
```

Appendix A-7

function Depth = ProfileMatch(profile3, LUT)

```
% Input parameters are 'profile3' – the normalised profile of a defocused edge under
% investigation, and 'LUT' – the 2-D look-up table of edge profile depth 'standards'.
% The size of the look-up table is determined, and a loop matching the number of
% table entries is implemented.  Each iteration of the loop compares the bounded set
% of test profile elements against the same set of elements for the current look-up
% table profile – as determined by the loop number iteration.  The difference between
% the sum of these two sets produces Residual 1.  Within the same loop iteration, the
% test profile set is also compared to the next profile in the look-up table.  This
% enables the comparison in each loop iteration to determine whether the current
% standard comparison (LUT(i)) is a better match than LUT(i+1) – if it is the best
% match is found and the loop is terminated.

% W. E. CROFTS, Image Processing & Expert Systems Group, University of Warwick

 [R C]=size(LUT); %R = No of depth profile entries in LUT

% Check each entry in look-up table LUT
%Note: If current profile (profile3) is 'higher' up the table (LUT) than
%the table position being compared (LUT(i,:) then the 'Residual' generated
%is positive.

%ProfileMatch2 restricts the sum of elements being analysed to those
%between columns 30 to 50 inclusive.  This represents the point where the
%profiles display the maximum differentiation.

for i = 1:(R-1)                 %Range set to R-1 otherwise LUT(i+1) goes out of range
   Residual1 = sum(LUT(i,30:50)) - sum(profile3(1,30:50));
   Residual2 = sum(LUT(i+1,30:50)) - sum(profile3(1,30:50));
   SumRes(i)=Residual1;
   if Residual1 <= 0
      Depth = i;                %Fit exact match to min value or,
      break                     %fit outlier (off 'bottom' of table) to min value
   elseif Residual2 = = 0
      Depth = i + 1;            %Best depth match = i+1 in LUT
      break
   elseif (i+1==R) & (Residual2 > 0)
      Depth = i + 1;            %Fit outlier (off 'top' of table) to max value
      break
   elseif (Residual1 > 0) & (Residual2 > 0)
      continue
   elseif (Residual1 > 0) & (Residual2 < 0)
      if (abs(Residual1) > abs(Residual2))
         Depth = i+1;           %Best depth match = i+1 in LUT
         break
      else
         Depth = i;             %Best depth match = i in LUT
         break
      end
   end
end
```

Appendix A-8

This program extracts and analyses defocused edge-profiles from an image of a projected structured light pattern on a planar screen. A previously produced 2-Dimensional look-up table containing typical edge profiles as depth 'standards' is loaded into the program, and is used in a profile matching algorithm in order to estimate the depth.

```
% ImageDepthCal.m  -  Extracts edge profiles from a series of
% images of spatially-shifted projected structured-light patterns. this
% code is intended to analyse SINGLE DEPTH PLANAR IMAGES.
% Each edge profile is matched to a table (LUT) of stored depth profiles
% generated by AverageCalc.m. The corresponding profile, and hence depth,
% is determined by calculating the minimum sum of residuals between
% the extracted profile and those in LUT.

% W. E. CROFTS, Image Processing & Expert Systems Group, University of Warwick

clear                   % Clear workspace
load 'LookUpTable3'     % Load edge profile look-up table


disp(' ')
disp('  **  PROGRAM TO ANALYSE DEPTH IN A FULL IMAGE  **')
disp(' ')
Gate = input('Load an image? Enter 1 to continue or 0 to end > ');
disp(' ')

Im = 1; % Set image counter
while Gate == 1         % User control for number of images to be analysed

   %%LOAD IMAGE VIA GUI WINDOW%%
   [infile, pathname]=uigetfile('\Research\PhD\MATLABfiles\structuredlight\
    *.*','Please select a file');
   %UIGETFILE - Interactively retrieves a filename into 'infile' by displaying a dialog box.

   infile1 = [pathname infile];
         %Associates full path and filename to INFILE1

   [image1,map] = imread(infile1);
         %[X,MAP] = IMREAD(FILENAME,FMT) reads the indexed image in FILENAME
         %into X and its associated colormap into MAP. Colormap values in the
         %image file are automatically rescaled into the range [0,1].


   %% EXTRACT ALL EDGE CURVE SAMPLES ON EACH ROW %%

   [M,N] = size(image1);        %M = no of rows, N = no of columns
   DepthDefault = 0;            %Initialise default depth for DepthMap



   for m = 1:M                  % Sample rows in image
      profile1=(image1(m, 1:N)); % Creates sample from selected row
```

```
profile1=smooth(double(profile1));          %Smooth 'profile1' to eradicate 'glitches'
ave=((max(profile1) + min(profile1))/2);
%Finds average intensity value, i.e. half profile height

%Identifies first 'rising' edge in row
n = 1;                          % Column counter to start of row
if profile1(n) >= ave           %If first pixel is WITHIN WHITE STRIPE
   % Find position of first FULL 'rising' edge to right of black stripe
   while (profile1(n) >= ave) & (n < N)          %Is current position still within
                                                 %white stripe?
      n = n + 1;                %Move to right to find end of white stripe
   end
   while (profile1(n) < ave) & (n < N)           %Is current position still within
                                                 %black stripe?
      n = n + 1;                %Move to right to find end of black stripe
   end
   U1 = n;                       % Note position of first full rising edge

elseif profile1(n) < ave        %If first pixel is WITHIN BLACK STRIPE
   % Find position of first FULL 'rising' edge to right of black stripe
   while (profile1(n) < ave) & (n < N)           %Is current position still within
                                                 %black stripe?
      n = n + 1;                %Move to right to find end of black stripe
   end
   while (profile1(n) >= ave) & (n < N)           %Is current position still within
                                                 %white stripe?
      n = n + 1;                %Move to right to find end of white stripe
   end
   while (profile1(n) < ave) & (n < N)           %Is current position still within
                                                 %black stripe?
      n = n + 1;
   end
   U1 = n;                       %Note position of first full rising edge
end

%Identifies last 'rising' edge in row
n = N;                          %Column counter to end of row
if profile1(n) >= ave           %If last pixel is WITHIN WHITE STRIPE
   % Find position of last FULL 'falling' edge to left of white stripe
   while (profile1(n) >= ave) & (n > 1)          %Is current position still within
                                                 %white stripe?
      n = n - 1;                %Move to left to find start of white stripe
   end
   while (profile1(n) < ave) & (n > 1)           %Is current position still within
                                                 %black stripe
      n = n - 1;                %Move to left to find start of black stripe
   end
   while (profile1(n) >= ave) & (n > 1)
   n = n - 1;
   end
   UL = n;                       % Note position of last full rising edge
elseif profile1(n) < ave        %If last pixel is WITHIN BLACK STRIPE
   % Find position of last FULL 'rising' edge to left of black stripe
```

```
    while (profile1(n) < ave) & (n > 1)          %Is current position still within
                                                 %black stripe?
      n = n - 1;            %Move to left to find start of black stripe
    end
    while (profile1(n) >= ave) & (n > 1)         %Is current position still within
                                                 %white stripe?
      n = n - 1;            %Move to left to find start of white stripe
    end
    UL = n;                 %Note position of last full rising edge
  end

  edgepos = U1;
  while edgepos <= UL
    [profile2,EndCurve] = RisingCurve(edgepos,ave,N,profile1);
    X1 = length(profile2);
    if rem ((140-X1),2) == 0              %If 'even' no of elements in profile
      Subzeros = zeros(1,(140-X1)/2);     %Set vector of 'zeros' for shift mask
      Subones = ones(1, (140-X1)/2);      %Set vector of 'ones' for shift mask
    else                                  % If 'odd' no of elements in profile
      Subzeros = zeros(1,(140-X1+1)/2);   %Set vector of 'zeros' for shift mask
      Subones = ones(1, ((140-X1+1)/2)-1); %Set vector of 'ones' for shift mask
    end
    NewXframe = [Subzeros profile2 Subones];      % Increase vector length to 140

    %Shift profile to locate 'EndCurve' at right-hand end of vector
    Shift = length(Subones);                      %Determine required 'right' shift
    profile3 = circshift(NewXframe,[0,Shift]);    %Shift 'EndCurve' to rightmost end
    profile3 = profile3(41:140);                  %Remove extraneous elements

    Depth = ProfileMatch2(profile3, LUT);

    %% Place depth estimate in current Depth Map
    %Currently 'm' is row position, 'edgepos' is col position
    DepthMap(m, edgepos) = Depth;
    NextEdge = Riseseek(edgepos,ave,profile1,N);
    edgepos = NextEdge;
  end
end

infile2=num2str(infile);
disp(' ')
disp(['Image DepthMap for image ',infile2,' completed'])
disp(' ')
disp('Do you wish to see error statistics?')
disp(' ')
disp('This will show individual image stats for shifted pattern series')
disp('or for discrete images a statistics plot can also be generated')
disp(' ')
Er = input('Enter 1 to compute error, or 0 to end > ');
disp(' ')

%%Calculate error statistics for discrete depth plain%%

if Er == 1                  %Extract Depth map values into vector
  i=1;                      %Intialise index
```

```
      for X = 1:M
         for Y = 1:N
            if DepthMap(X,Y) > 0.2
               DepthVector(i) = DepthMap(X,Y);
               i=i+1;
            end
         end
      end
      %Calculate mean & adjust to actual depth (rather than table index)
      Mean = num2str(mean(DepthVector)+ 89);
      Std = num2str(std(DepthVector));
      if length(infile)==8
         Dist=str2num(infile(1:2));
      else
         Dist=str2num(infile(1:3));
      end

      LowRes1 = (min(DepthVector)+89) - (mean(DepthVector)+89);
      LowRes = num2str(LowRes1);
      HighRes1 = (max(DepthVector)+89) - (mean(DepthVector)+89);
      HighRes = num2str(HighRes1);
      Error = num2str((((mean(DepthVector)+89) - Dist)/Dist)*100);
      Dist2 = num2str(Dist);
      Min = num2str(min(DepthVector)+89);
      Max = num2str(max(DepthVector)+89);
      EstValue = num2str(round(mean(DepthVector)+89));
      disp(['Image analysed is ',infile2])
      disp(['The known depth is ',Dist2,' cm'])
      disp(['The minimum estimated depth value = ',Min,' cm'])
      disp(['The maximum estimated depth value = ',Max,' cm'])
      disp(['The mean of calculated depth values = ', Mean,' cm'])
      disp(['The standard deviation of calculated depth values = ',Std])
      disp(['The ESTIMATED DEPTH VALUE = ',EstValue,' cm'])
      disp(['The estimated residuals = ',LowRes,' cm to ',HighRes,' cm'])
      disp(['The ERROR = ',Error,'%'])

   else
      disp(' ')
      disp('You have chosen not to calculate error data for this image ')
      disp('therefore it will not be included in the statistics plot.')
      disp('If you want it to be included, you will have to reload it ')
      disp('as the next image and choose to compute the error.')
   end

   disp(' ')
   disp(['Image ',infile2,' has been analysed'])
   Gate = input('Load another image? Enter 1 to continue or 0 to end > ');
   disp(' ')

   Im = Im + 1;
   clear DepthVector DepthMap
   disp(' ')

end
```

```
disp(' ')
disp('Do you wish to generate a statistical error plot(fixed Depth Plane only)?')
ErrPlt = input('Enter "1" to generate a plot or "0" to end > ');
if ErrPlt = = 1
   errorbar(DistTable,MeanTable,LowResTable,HighResTable);
   hold on;
   plot(DistTable,DistTable,'--r');
end

disp(' ')
disp('Do you wish to plot the Depth Map profile? ')
PlotMap = input('Enter "1" to plot or "0" to continue > ');

if PlotMap = = 1

   % Specify the interpolation step size
   row_step = 10;
   col_step = 10;
   [InterpDepthMap] = PlotDepthMap(DepthMap, row_step, col_step, DepthDefault);

end

%**************************************************************************
%


fclose ('all');
%FCLOSE  -  Close a file.
% FCLOSE('all') closes all open files, except 0, 1 and 2.
```

Appendix A-9

```
% FourD-LUT-Generator.m  -  An M-file to extract a number of edge profiles
% from calibration images.
% The profiles will be extracted from designated windows at each
% depth, and will be averaged to produce a 'best estimate' profile for each
% window at each depth. Hence, the final Look-up Table (LUT) will be a
% 4-Dimensional array of M x N standard depth profiles per image for each
% of the depths imaged in the calibration process.
% Edge profiles for specific depths were generated using a vertical stripe
% projected light pattern.  This pattern was moved laterally
% (left-to-right) across the object in order to increase the spatial
% resolution of the depth estimation across the image. The pattern can be
% moved 'n' times per pattern period, and the depth object is therefore
% imaged 'n' times per depth.  Each of the 'n' depth images will need
% to be analysed to produce the 3-Dimensional LUT array that represents
% each depth within the overall 4-Dimensional array.

% W. E. CROFTS
% Image Processing & Expert Systems Group, University of Warwick
% Ver 1:  29th January, 2007
% Ver 2:  8th February, 2007
% Ver 3:  19th March, 2007
clear

disp(' ')
disp('Program to Generate a 4-Dimensional Lookup Table from Calibration Images')
disp(' ')
disp('Enter the number of DEPTH image-sets (1cm intervals) in the calibration set')
disp('Note: This does not mean the number of images taken at each depth')
disp('to capture the projected pattern movement.')
disp('e.g. Depth range of 90cm to 106cm = 17 depth image-sets at 1cm intervals')
disp(' ')
NoOfDepths = input('Enter number of DEPTH image sets required > ');
disp(' ')
disp('The number of images per depth is the no of pattern movements in cycle')
NoOfPatterns = input('Enter number of images per depth > ');
disp(' ')
disp('Enter a value to represent the vertical resolution (rows per LUT profile)')
disp('that you wish the Lookup Table to have.')
disp('i.e. If you wish to generate a window 1/200th of a 1200 row image')
disp('     Note: This would generate a standard profile for each set of 6 rows')
RowRes = input('then enter "200", etc. > ');
disp(' ')
disp('By inspection of the row profile of the images to be used to')
disp('generate the LUT.')
ExpectedEdgeCount = input('State the number of VALID edge profiles per row > ');
disp(' ')

%% CREATE EMPTY ARRAY FOR 4-D LUT STANDARD PROFILES %%
```

```
% This 4D lookup table is organised as
% (No of Row Blocks x No of Profiles x No of depths x profile)
% Note: Standard Array profiles will be adjusted to give 10 profiles per
% pattern shift
% i.e. For 20 shifted patterns gives
% 10 x 20 = 200 shifted profiles across each image row
%LookUp_Table_4D(RowRes,64,NoOfDepths,1:100)=zeros;

%% ANALYSE DEPTH IMAGES TO GENERATE 4D LOOKUP TABLE %%
NoOfPatterns2 = num2str(NoOfPatterns);
disp(['Analysing ',NoOfPatterns2,' images for first depth?'])
DepthSet = 1;            % Initialise counter for set of depth images

while DepthSet <= NoOfDepths
   for Im = 1:NoOfPatterns
      %%LOAD IMAGE VIA GUI WINDOW%%
      [image1, infile] = ImageLoadFunc(Im,DepthSet);
      %% STORE IMAGES FROM DEPTH SET %%
      [M,N] = size(image1);            %M = row, N = column
      [P1,PL] = ProfilePosLocateFunc(image1,M/2,N);
      DepthSetOrder(Im) = P1;          %Position of first profile in image
      ImageStore(:,:,Im) = image1;     %Store 'NoOfPattern' images in array
   end
   %%FIND LEFT-TO-RIGHT ORDER OF IMAGE PATTERN MOVEMENTS%%
   DepthSetReordered = SortDepthSetFunc(DepthSetOrder,NoOfPatterns);
   %DepthSetReordered returns pattern positions in the correct
   %left-to-right order.  This is a 2-D array in the format:
   %(First profile position, Image No)
   clear DepthSetOrder

   for Im = 1:NoOfPatterns
      ImageNo=Im
      %Extract images from store in left-to-right order
      CurrentImage = ImageStore(:,:,DepthSetReordered(2,Im));
      RowSample = M/RowRes;        %No. of rows to form averaged profile
      RowStart = 1;                %Initialise start row for image analysis
      RowEnd = (M - (RowSample - 1));
      while RowStart <= RowEnd
         % Find Row 'start and end' profile locations
         for Row = RowStart:RowSample
            [P1,PL,profile1,ave] = ProfilePosLocateFunc(CurrentImage,Row,N);
            edgepos = P1;
            edgecount = 0;
            R=mod((Row-1),(M/RowRes))+1;          %modulus to get correct
                                                  %row count in vertical block
            while edgepos <= PL
               % Generate normalised edge profiles for Row
               [profile2] = RisingCurve(edgepos,ave,N,profile1);
               %Shift and Scale the profile
               [profile3] = ShiftnScaleFunc(profile2);
```

```
            edgecount = edgecount + 1;
            %% ** TEMP FOR TESTING START**
            if ((Row==950) && (edgepos>=150) && (edgepos<=700))
               Value = sum(profile3(30:50));
               Value2 = num2str(Value); Row2 = num2str(Row);
               Im2 = num2str(Im);
               %disp(['Row ',Row2,', image ',Im2,', has first profile value of
               %',Value2]);
               FirstProMatrix(Im,edgecount)=Value;
            end
            %% ** TEMP FOR TESTING END**
            ProfileArray(:,R,edgecount) = profile3;
            NextEdge = Riseseek(edgepos,ave,profile1,N);
            edgepos = NextEdge;
         end
         if edgecount ~= ExpectedEdgeCount
               [ProfileArray] =
               ProfArrayCorrectFunc(ProfileArray,edgecount,ExpectedEdgeCount);
         end
      end
      %Average for profile positions per row block
      for EdgeNum = 1:ExpectedEdgeCount %For each edge profile
         for RowNum = 1:(M/RowRes)              %Each row in block
            if RowNum == 1
               ProfileAccum = ProfileArray(:,RowNum,EdgeNum);
            else
               ProfileAccum = ((ProfileAccum +
               ProfileArray(:,RowNum,EdgeNum))/2);
            end
         end
         %Save averaged column profile in 3-D look-up table
         RowPos = (RowSample/(M/RowRes));          %Row block number
         ColPos = (((EdgeNum - 1) * NoOfPatterns) + Im);     %Profile Column
                                                             %number

         LookUp_Table_4D(:,RowPos,ColPos,DepthSet)=sum(ProfileAccum(30:5
         0));
      end
      %Increment 'RowStart' and 'RowSample'
      RowStart = RowStart + (M/RowRes);
      RowSample = RowSample + (M/RowRes);
      clear ProfileArray ProfileAccum
   end
end
%% ** TEMP FOR TESTING START**
[E,I] = size(FirstProMatrix)
i=0;
for r=1:E
   for c=1:I
      i=i+1;
```

```
            FirstProVector(i)=FirstProMatrix(r,c);
        end
    end
    %% ** TEMP FOR TESTING END**
    DepthSet = DepthSet + 1;
    disp(' ')
    disp(['Image ' infile ' has been analysed and stored in LUT'])
    disp(' ')
    if Im < NoOfPatterns
        disp('ANALYSING NEXT IMAGE ')
    end
    disp(' ')
end

LookUp_Table_4D = ModFunc(LookUp_Table_4D);
save('LUT_4D','LookUp_Table_4D')

disp('4D Lookup Table has been generated and saved')
%*****************************************************************

%*****************************************************************



fclose ('all');
        %FCLOSE  -  Close a file.
        % FCLOSE('all') closes all open files, except 0, 1 and 2.
```

Appendix A-10

```
% DfDMovingPatternAnalysisV3.m  -  Extracts edge profiles from a series of
% images of spatially-shifted projected structured-light patterns.
% Each edge profile is matched to a table (LookUp_Table_4D) of stored
% depth profiles generated by FourD_LUT_Generator.m. The row and column
% position (i.e. row and pattern edge profile no.) of the profile under
% analysis is used to determine which 'window' of the 4D LUT will be
% investigated in order to determine the best depth estimate for that
% spatial position in the image.


% W. E. CROFTS, Image Processing & Expert Systems Group, University of
%Warwick
% Ver 1:  21st February 2007
% Ver 2:  19th March 2007 - increased (200x200) resolution LUT
% Ver 3:  18th April 2007 - neutralise image texture effects by normalising
% intensity to mean of original image.

clear                   % Clear workspace
load 'LUT_4D'           % Load 4-Dimensional edge profile look-up table
% Note: Table matrix is (100 x 200 x 200 x 17)
% This represents (profile elements x row block x profile column x depth)
% 'Depth' in the original version of 4D-LUT is 90cm to 106cm inclusive
% at 1cm intervals


disp(' ')
disp('  **  PROGRAM TO ESTIMATE DEPTH IN A PROJECTED
STRUCTURED-LIGHT IMAGE  **')
disp(' ')
NoOfImages = input('Enter number of images per cycle of pattern movement > ');
disp(' ')
NoOfEdges = input('Enter expected number of edge profiles per row > ');
disp(' ')

Start = input('Load application image set? Enter 1 to continue or 0 to end > ');
if Start == 0
   disp(' ')
   disp('PROGRAM TERMINATED')
   return
end


%%************************************************************
%% LOAD & STORE APPLICATION IMAGE SET FOR ANALYSIS %%
%%************************************************************

% LOAD NORMAL IMAGE TO ESTABLISH INTENSITY MEAN
DepthSet=-1;  % Only req'd to comply with ImageLoadFunc input parameters
```

```
disp(' ')
disp('Load normal (plain) image to establish intensity mean')
Imload = -1;                    % Set to initiate 'plain' image in ImageLoadFunc
[imageP, infile] = ImageLoadFunc(Imload,DepthSet);
IntMean = mean(imageP(:));  % Find mean of original image
[V,W] = size(imageP);           %V = rows, W = columns
imageP=double(imageP);      % image type to 'double' for correct multiplier value
%Set up Multiplier array to neutralise image texture
for Vi = 1:V
   for Wi = 1:W
      Multiplier(Vi,Wi) = IntMean/(imageP(Vi,Wi));
      % Try effect on original image
      ImageCheck(Vi,Wi) = Multiplier(Vi,Wi)*imageP(Vi,Wi);
   end
end
disp(' ')
disp('Now load remaining application images')
for Imload = 1:NoOfImages
   %% Load Image via GUI Window %%
   [imageApp, infile] = ImageLoadFunc(Imload,DepthSet);
   %% NOTE PROJECTED PATTERN POSITION  %%
   [P1,PL] = ProfilePosLocateFunc(imageApp,V/2,W);
   ImageSetOrder(Imload) = P1;       %Position of first profile in image
   %% NEUTRALISE ORIGINAL IMAGE TEXTURE %%
   imageApp=double(imageApp);     % Modify type for next calculation
   for Vi = 1:V
      for Wi = 1:W
         image1(Vi,Wi) = Multiplier(Vi,Wi)* imageApp(Vi,Wi);
      end
   end
   %% STORE IMAGE FOR DfD PROCESSING %%
   ImageStore(:,:,Imload) = image1;            %Store 'NoOfPattern' images in array
end

%%FIND LEFT-TO-RIGHT ORDER OF IMAGE PATTERN MOVEMENTS%%
ImageSetReordered = SortDepthSetFunc(ImageSetOrder,NoOfImages);
%DepthSetReordered returns pattern positions in the correct
%left-to-right order.  This is a 2-D array in the format:
%(First profile position, Image No)
clear ImageSetOrder

disp(' ')

%FIX VERTICAL & HORIZONTAL RESOLUTION FOR ANALYSIS VIA 4D-LUT %
[P,R,C,N]=size(LookUp_Table_4D);
RowRes = R;             % No of vertical divisions to which image must be divided
RowBlock = V/R;      % Sets no of rows in depth estimate 'window'
ColRes = C;             % No of horizontal divisions to which image must be divided
ColBlock = W/C;      % Sets no of columns in depth estimate 'window'
```

```
%% SET-UP DEPTHMAP %%
DepthDefault = 20;                      %Initialise default depth for DepthMap
DepthMap(1:V,1:W) = DepthDefault;       %Initialise DepthMap to default depth
TotalCount = 0;                         %Initialise counter for total no of edges
                                        %analysed

for Im = 1:NoOfImages       % User control for number of images to be analysed

   %Extract images from store in left-to-right order
   CurrentImage = ImageStore(:,:,ImageSetReordered(2,Im));

   %% IMAGE ANALYSIS %%
   for m = 1:V              % Sample rows in image
      % Fix RowNumber in which this row will be analysed in 4D-LUT
      RowNumber = floor((m-1)/RowBlock)+1;
      %Identify first & last valid 'rising' edge in row
      [P1,PL,profile1,ave] = ProfilePosLocateFunc(CurrentImage,m,W);
      edgepos = P1;
      edgecount = 0;        % Initialise edge counter
      i=0;                  % Loop counter for test section below
      while (edgepos <= (PL+3)) & (edgecount <=9)
         % Note: '+3' allows for slight variation in PL when row is analysed
         edgecount = edgecount + 1;
         if edgecount > NoOfEdges     % Will produce 'out of range' error
            m2 = num2str(m);
            Im2 = num2str(Im);
             disp(['Out of Range Error in Number of edges on row ',m2,' of image
            ',Im2])
            disp(' ')
            break           % in ProfileMatchFunction
         end
         % Generate normalised edge profile for row
         [profile2] = RisingCurve(edgepos,ave,W,profile1);
         % Shift and Scale the profile
         [profile3] = ShiftnScaleFunc(profile2);
         % Fix ColNumber in which this profile will be analysed in 4D-LUT

         %% ** TEMP FOR TESTING - START**
         if ((m==950) & (edgepos>150) & (edgepos<700))
            i=i+1;
            Value = sum(profile3(30:50));
            Value2 = num2str(Value); Row2 = num2str(m);
            Im2 = num2str(Im);
            %disp(['Row ',Row2,', image ',Im2,', has first profile value of ',Value2]);
            FirstProMatrix(Im,i)=Value;
         end
         %% ** TEMP FOR TESTING - END**

         ColNumber = (((edgecount-1)* NoOfImages)+Im);
         % Determine best depth estimate from 4D-LUT
```

```
        Depth = ProfileMatchFunc(profile3, RowNumber, ColNumber, N,
DepthDefault,LookUp_Table_4D);

        %% Place depth estimate in current Depth Map
        %Currently 'm' is row position, 'edgepos' is col position
        %The 'ColBlock' position relative to current 'edgepos' will be filled
        %with the estimated depth value. Hence....
          %ColBlockEnd = ColBlock * ColNumber;
          %ColBlockStart = ColBlockEnd - ColBlock + 1;
          %DepthMap(m, ColBlockStart:ColBlockEnd) = Depth;
        DepthMap(m, edgepos) = Depth;
        NextEdge = Riseseek(edgepos,ave,profile1,W);
        edgepos = NextEdge;
      end
   end
end

%% ** TEMP FOR TESTING START**
[E,I] = size(FirstProMatrix)
i=0;
for r=1:E
   for c=1:I
      i=i+1;
      FirstProVector(i)=FirstProMatrix(r,c);
   end
end
%% ** TEMP FOR TESTING END**

[DepthMap] = DepthMapInterpFunc(DepthMap, DepthDefault);
clear ImageStore
infile2=num2str(infile);
disp(' ')
disp(['DEPTH MAP FOR IMAGE ',infile2,' COMPLETED'])
disp(' ')
beep on
beep

disp(' ')
disp('Do you wish to calculate the error statistics for this image')
disp('NOTE: This function is for SINGLE-DEPTH PLANAR test images only')
ErrorStart = input('Enter "1" to calculate error stats or "0" to end > ');
if ErrorStart = = 1
   ErrorCalcFunc(DepthMap, V, W, ColBlock, DepthDefault);
end
beep

disp(' ')
disp('Do you wish to plot the Depth Map profile?')
PlotMap = input('Enter "1" to plot or "0" to continue > ');
if PlotMap = = 1
```

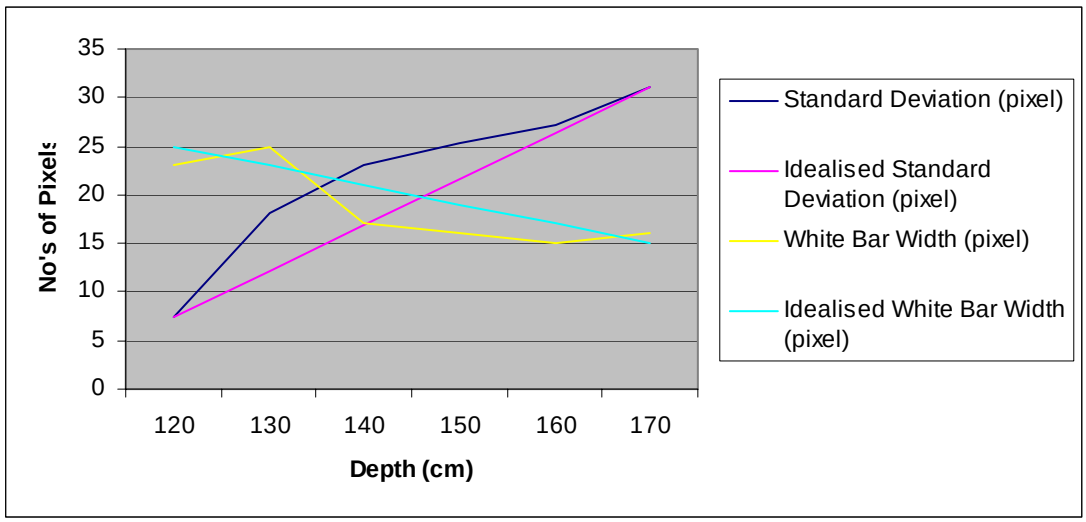
```
   % Specify the interpolation step size
   row_step = 10;
   col_step = 10;
   [InterpDepthMap] = PlotDepthMap(DepthMap, row_step, col_step,
DepthDefault);
end
beep



%*****************************************************************

%*****************************************************************



fclose ('all');
        %FCLOSE  -  Close a file.
        % FCLOSE('all') closes all open files, except 0, 1 and 2.
```

Appendix B-1

Experiment 1: Depth cue variation versus depth

1) Table and resultant plot of the variation of Standard Deviation and Width of White Bar versus Depth

Depth (cm)	Standard Deviation (pixel)	Idealised Standard Deviation (pixel)	White Bar Width (pixel)	Idealised White Bar Width (pixel)
120	7.3877	7.3877	23	25
130	18.1741	12.1126	25	23
140	22.982	16.8375	17	21
150	25.2821	21.5623	16	19
160	27.1108	26.2872	15	17
170	31.0121	31.0121	16	15



The standard deviation values are calculated from the variation of light intensity over one period of the structured light pattern located in the centre of the image.

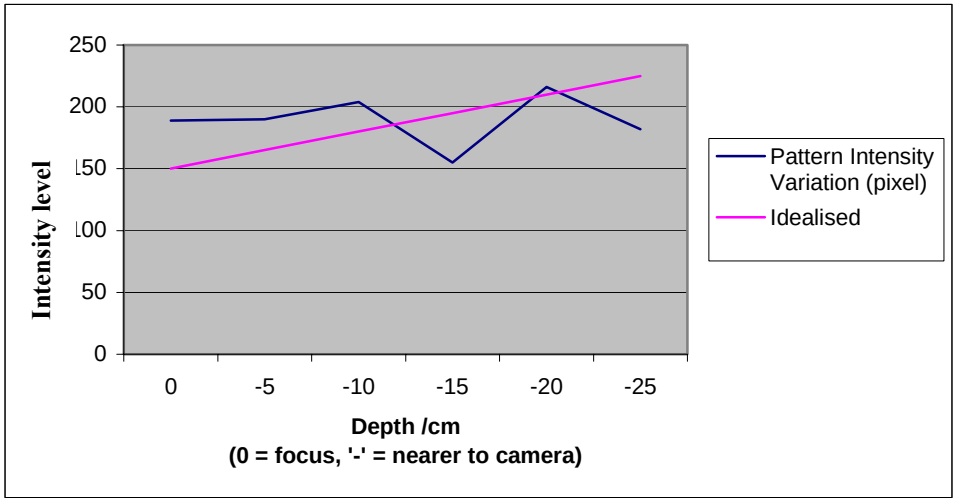
The white bar width values are calculated by determining the number of pixels that have a light intensity value at – or above – the calculated average for the light intensity profile in the image row of pixels under investigation.

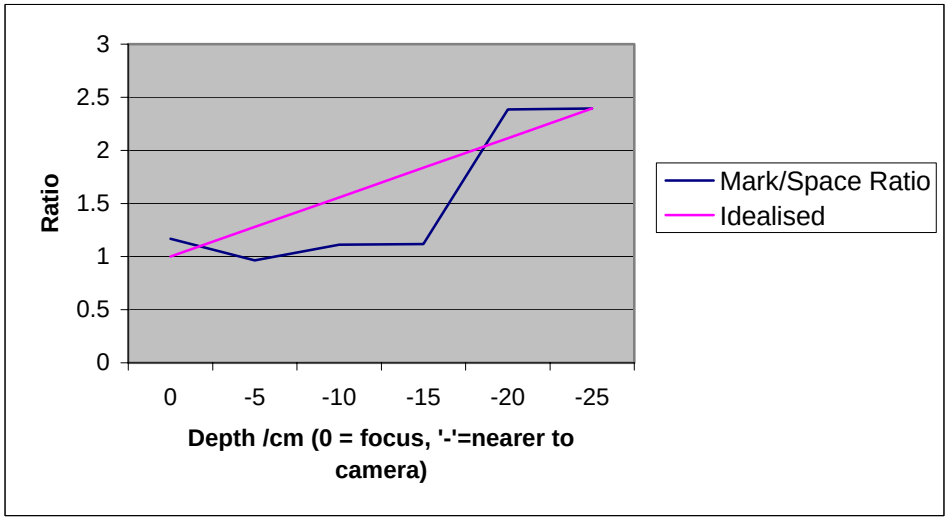
Both sets of values were determined by MATLAB programme 'VertStripeCalc.m' listed in Appendix A-2.

In both cases, the idealised values shown are for reference only. They are only to enable comparison with the form of linear response that would ideally be required in order to make the depth cue suitable for use in a depth-from-defocus system.

2) Table and resultant plot of the variation of Pattern Intensity Variation and Mark/Space Ratio (i.e. White Bar /Black Bar Width) versus Depth

Depth (cm)	Pattern Intensity Variation (pixel)	Idealised	Mark/Space Ratio @15% intensity	Idealised
0	189	150	1.1667	1
-5	190	165	0.96552	1.2786
-10	204	180	1.1111	1.5572
-15	155	195	1.12	1.8359
-20	216	210	2.3846	2.1145
-25	182	225	2.3931	2.3931





Note: The idealised plot for the Mark/Space ratio is particularly notional. When viewing Fig.4.1, it can be seen that the effect of moving the planar object screen (e.g. further away from the camera/projector) would cause the projected pattern image to 'spread' due to the divergent effect of the projection angle. However, the image of the pattern captured by the camera would tend to show a reduction in size, being further from the camera lens. These two effects would tend to cancel out, but to what extent is difficult to estimate. Hence, the notion of an ideal variation in mark/space ratio is mainly for reference, based on the values actually recorded.

Appendix B-2

Experiment 2: Investigation of Depth Cues in Projected Structured Light Patterns using a Foculus Firewire Camera

This experiment is discussed in detail in section 4.4 of chapter 4. The results recorded here show the response to a change in depth of four features (or depth cues) from the image of a projected light pattern. The depth range is from 170 cm (most in focus) to 115 cm (most out of focus) in steps of 5 cm, with the corresponding value for each depth cue extracted for each depth.

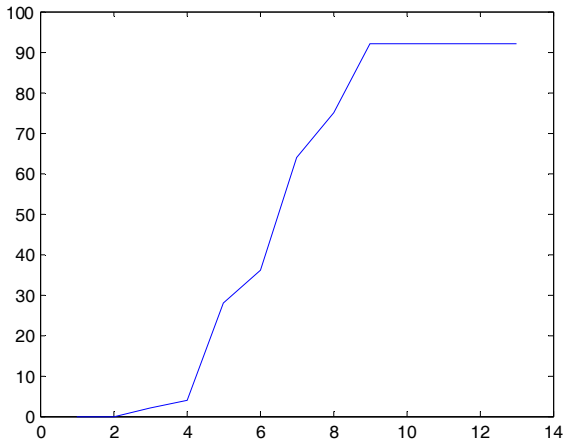
Also recorded for each depth position is a plot of the intensity profile of a sample 'rising' edge (i.e. black-to-white transition as the image row is scanned from left to right). The purpose of displaying this plot is to illustrate the changing form of the profile as depth varies.

A second plot displayed for each depth is that of a sample of a complete pattern period. This plot illustrates the change of pattern shape that affects the response of the remaining three depth cues (standard deviation of white bar pixels, pattern period, and mark/space ratio.

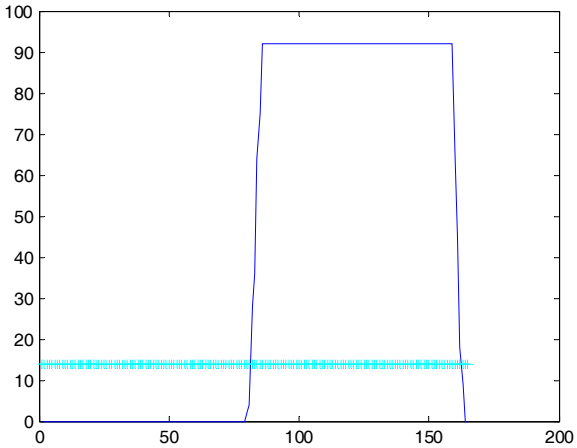
A table of recorded results is shown at the end of this appendix (pp B-9)

1) 170 cm from camera lens
Pulse height relative to minimum level = 92
Standard Deviation of extracted profile intensity levels = 10.943 (greyscale)
Period of pattern frequency in pixels = 165
Mark-Space Ratio at 15% of height intensity = 0.96429

Intensity profile of bar edge



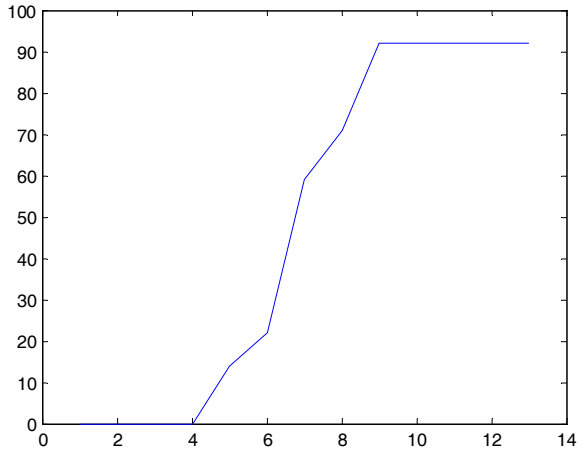
Full bar profile showing 15% intensity level (for MSR)



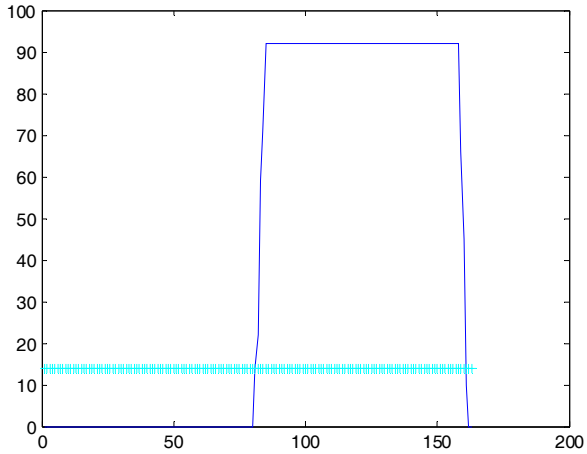
2) 165 cm from camera lens

Pulse height relative to minimum level = 92
Standard Deviation of extracted profile intensity levels = 11.0011 (greyscale)
Period of pattern frequency in pixels = 162
Mark-Space Ratio at 15% of height intensity = 0.96386

Intensity profile of bar edge



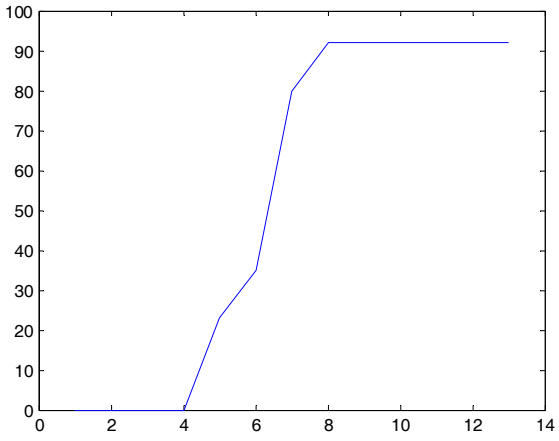
Full bar profile showing 15% intensity level (for MSR)



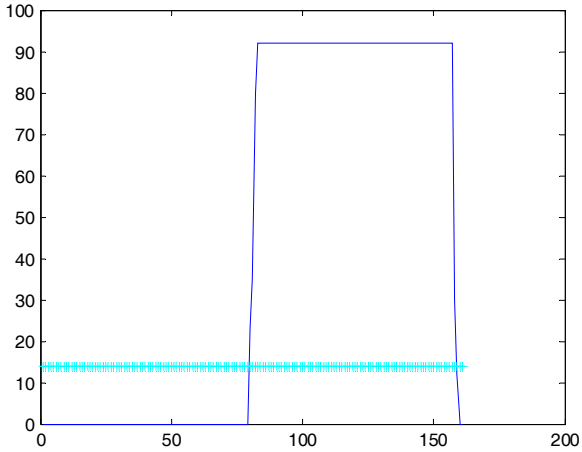
3) 160 cm from camera lens

Pulse height relative to minimum level = 92
Standard Deviation of extracted profile intensity levels = 11.0057 (greyscale)
Period of pattern frequency in pixels = 160
Mark-Space Ratio at 15% of height intensity = 0.96341

Intensity profile of bar edge



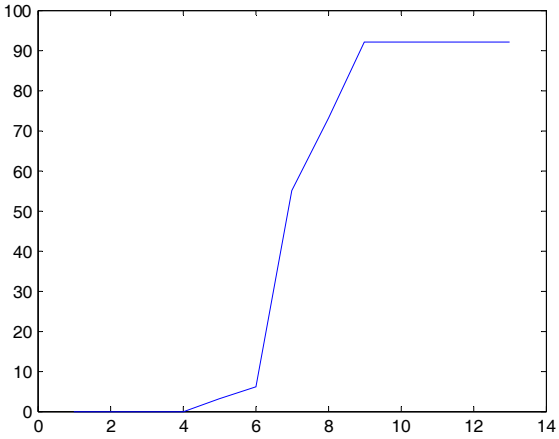
Full bar profile showing 15% intensity level (for MSR)



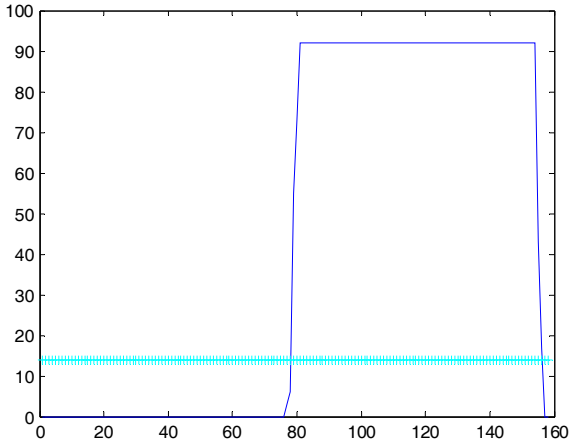
4) 155 cm from camera lens

Pulse height relative to minimum level = 92
Standard Deviation of extracted profile intensity levels = 11.0719 (greyscale)
Period of pattern frequency in pixels = 156
Mark-Space Ratio at 15% of height intensity = 0.975

Intensity profile of bar edge



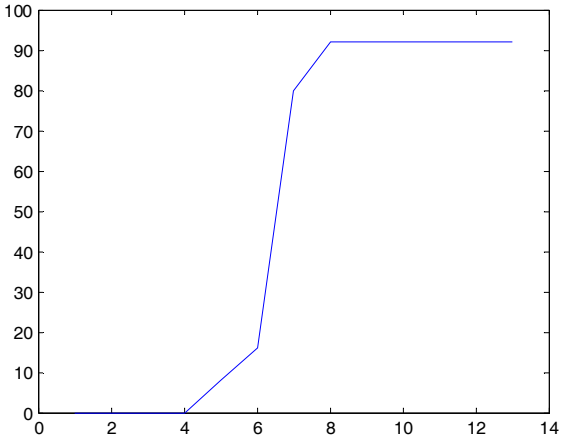
Full bar profile showing 15% intensity level (for MSR)



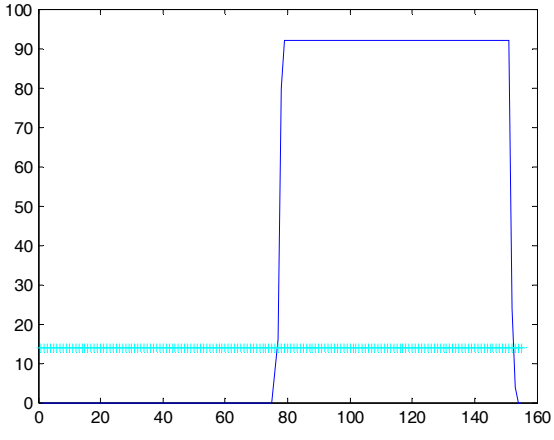
5) 150 cm from camera lens

Pulse height relative to minimum level = 92
Standard Deviation of extracted profile intensity levels = 11.0694 (greyscale)
Period of pattern frequency in pixels = 154
Mark-Space Ratio at 15% of height intensity = 0.9620

Intensity profile of bar edge



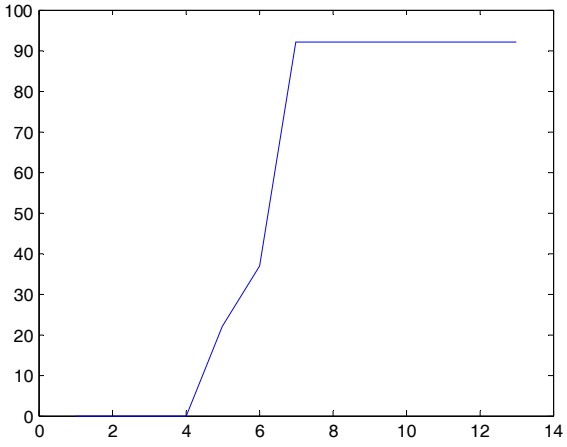
Full bar profile showing 15% intensity level (for MSR)



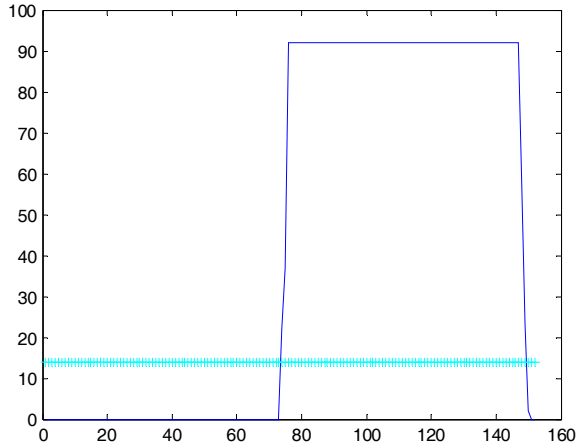
6) 145 cm from camera lens

Pulse height relative to minimum level = 92
Standard Deviation of extracted profile intensity levels = 11.0774 (greyscale)
Period of pattern frequency in pixels = 151
Mark-Space Ratio at 15% of height intensity = 1

Intensity profile of bar edge



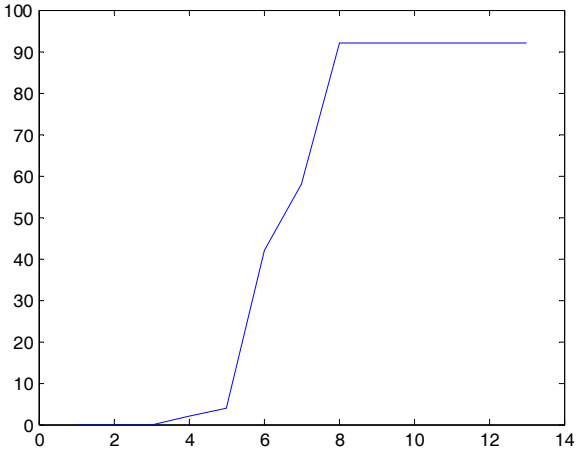
Full bar profile showing 15% intensity level (for MSR)



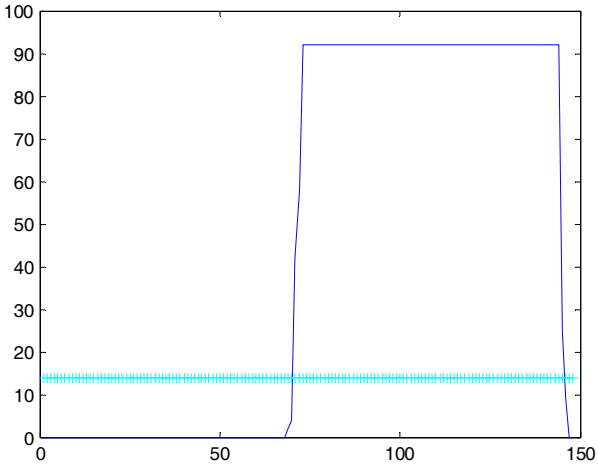
7) 140 cm from camera lens

Pulse height relative to minimum level = 92
Standard Deviation of extracted profile intensity levels = 11.2195 (greyscale)
Period of pattern frequency in pixels = 148
Mark-Space Ratio at 15% of height intensity = 1.0274

Intensity profile of bar edge



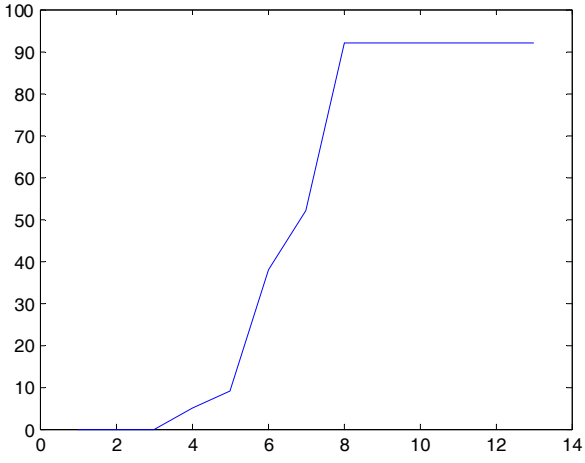
Full bar profile showing 15% intensity level (for MSR)



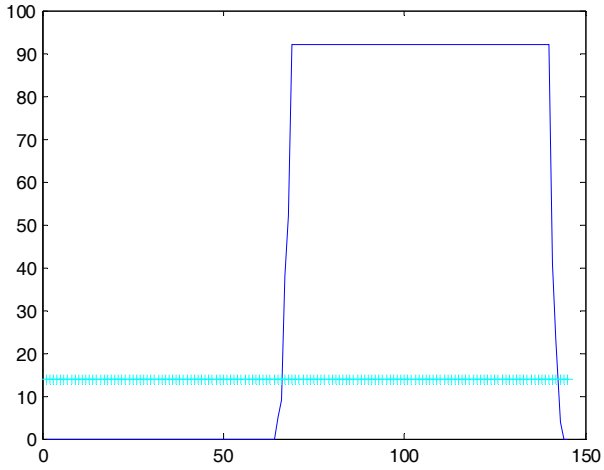
8) 135 cm from camera lens

Pulse height relative to minimum level = 92
Standard Deviation of extracted profile intensity levels = 11.2993 (greyscale)
Period of pattern frequency in pixels = 144
Mark-Space Ratio at 15% of height intensity = 1.1014

Intensity profile of bar edge



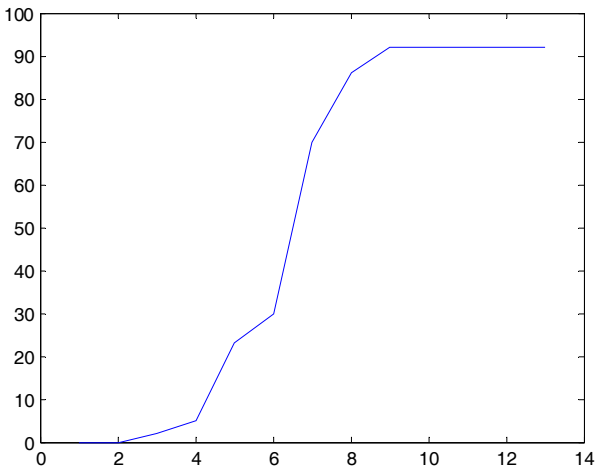
Full bar profile showing 15% intensity level (for MSR)



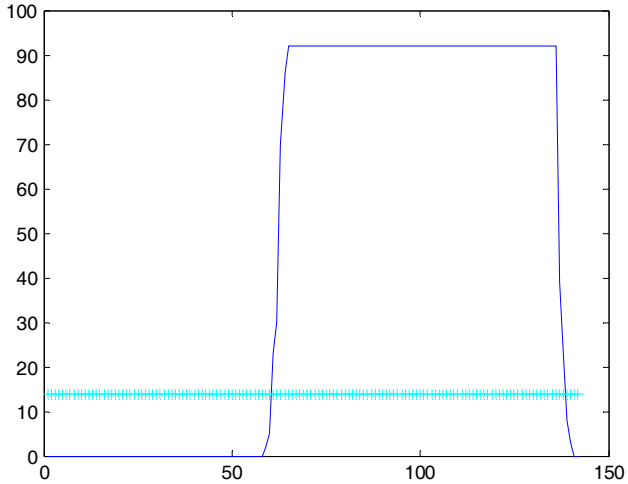
9) 130 cm from camera lens

Pulse height relative to minimum level = 92
Standard Deviation of extracted profile intensity levels = 11.5685 (greyscale)
Period of pattern frequency in pixels = 140
Mark-Space Ratio at 15% of height intensity = 1.2188

Intensity profile of bar edge



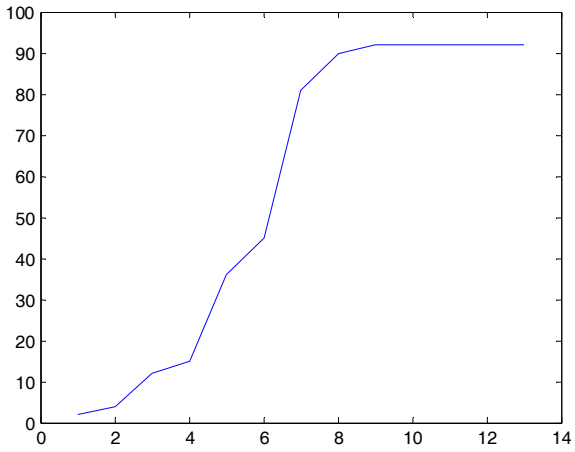
Full bar profile showing 15% intensity level (for MSR)



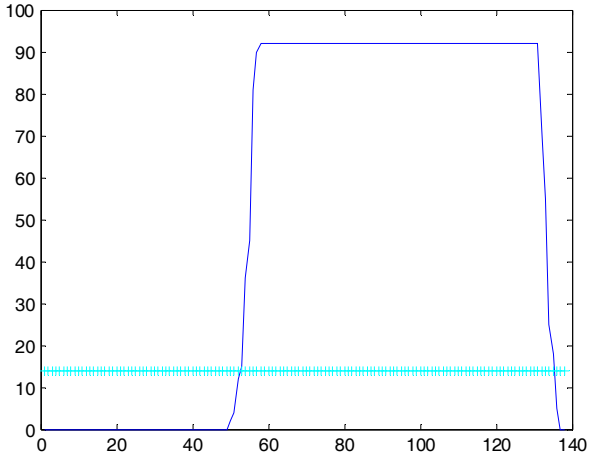
10) 125 cm from camera lens

Pulse height relative to minimum level = 92
Standard Deviation of extracted profile intensity levels = 11.9729 (greyscale)
Period of pattern frequency in pixels = 139
Mark-Space Ratio at 15% of height intensity = 1.5091

Intensity profile of bar edge



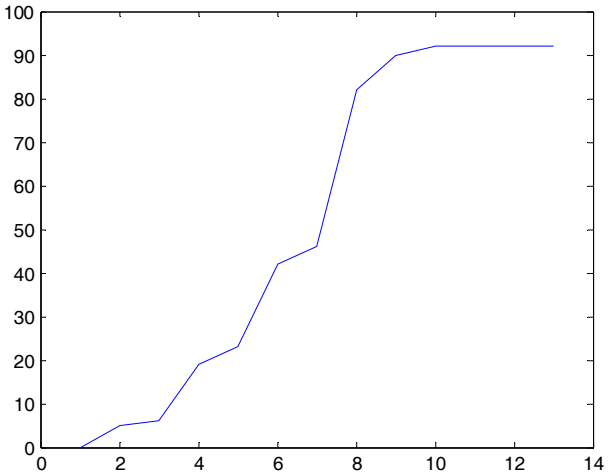
Full bar profile showing 15% intensity level (for MSR)



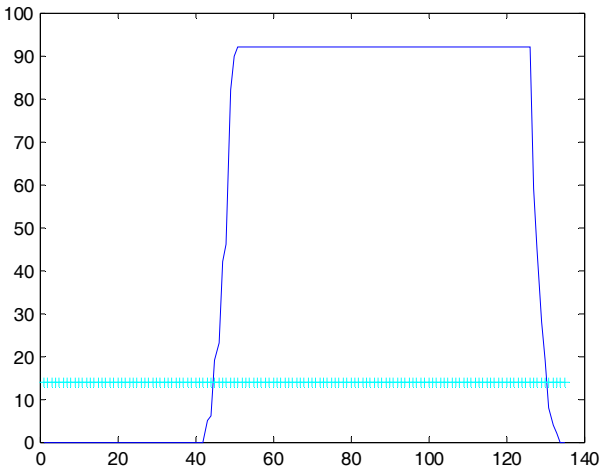
11) 120 cm from camera lens

Pulse height relative to minimum level = 92
Standard Deviation of extracted profile intensity levels = 12.1882 (greyscale)
Period of pattern frequency in pixels = 134
Mark-Space Ratio at 15% of height intensity = 1.7551

Intensity profile of bar edge



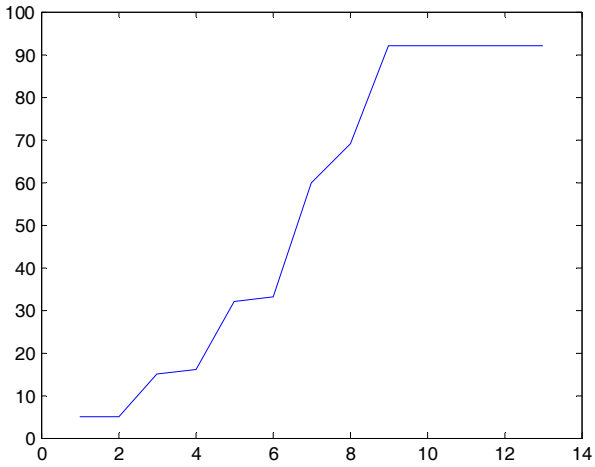
Full bar profile showing 15% intensity level (for MSR)



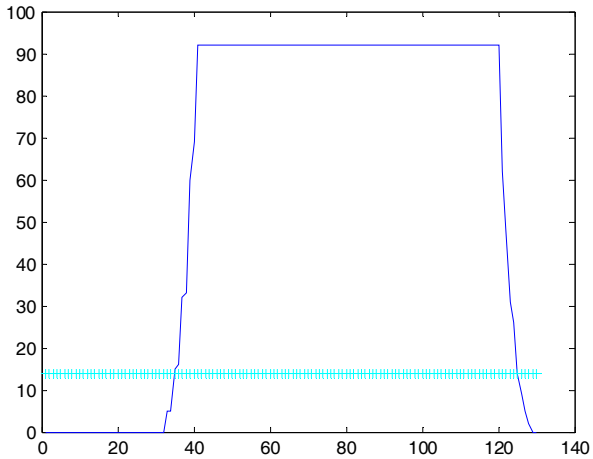
12) 115 cm from camera lens

Pulse height relative to minimum level = 92
Standard Deviation of extracted profile intensity levels = 12.6015 (greyscale)
Period of pattern frequency in pixels = 130
Mark-Space Ratio at 15% of height intensity = 2.3333

Intensity profile of bar edge



Full bar profile showing 15% intensity level (for MSR)



Investigation of Depth Cues in Projected Structured Light Patterns using Foculus Firewire Camera

Tabulated Results

Depth (cm)	Max Light Intensity Level (i.e. Step height) (0 to 255 Greyscale)	Standard Deviation of Extracted Profile (Greyscale)	Idealised	Period of Pattern Frequency (Pixels)	Idealised	Mark/Space Ratio (at 15% of intensity level)	Idealised
170	92	10.943	10.943	165	165	0.96429	0.245
165	92	11.0011	11.094	162	161.82	0.96386	0.396
160	92	11.0057	11.245	160	158.64	0.96341	0.547
155	92	11.0719	11.396	156	155.45	0.975	0.698
150	92	11.0694	11.547	154	152.27	0.962	0.849
145	92	11.0774	11.698	151	149.09	1	1
140	92	11.2195	11.849	148	145.91	1.0274	1.151
135	92	11.2993	12	144	142.73	1.1014	1.302
130	92	11.5685	12.151	140	139.55	1.2118	1.453
125	92	11.9729	12.302	139	136.36	1.5019	1.604
120	92	12.1882	12.453	134	133.18	1.7551	1.755
115	92	12.6015	12.604	130	130	2.3333	1.906

Appendix B-3

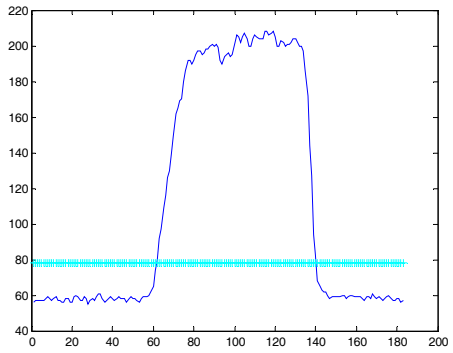
Experiment 3: The analysis of edge profile variation versus depth change

Analysing the use of Curve Fitting as a Depth Cue in Projected Structured Light Patterns using a high-resolution UEye Camera

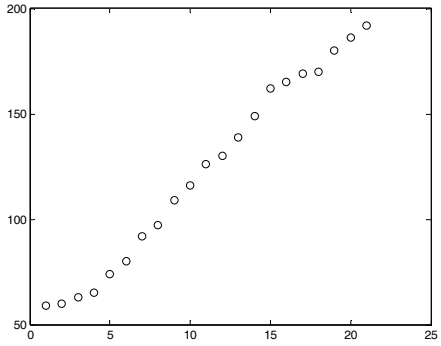
Camera used: UEye 1600 x 1200; 50mm lens; Focus fixed at 116 cm – the approximate position of focus of the projected light pattern (VertWhiteStripe02). A set of images of the vertical black/white bar pattern projected onto the movable planar screen was taken between depths of 115 cm (point of focus) and 80 cm, at 5 cm intervals. The 'spread' of the vertical white bar (and hence the spread of the bar edge profile) can easily be observed by eye as the defocus effects increase over the depth range of 115 cm to 80 cm.
By observation of Figure 4.14, the cubic polynomial equations do not offer any direct correlation between the actual depth and any aspect of the expressions themselves.

1) 115 cm from camera lens (point of pattern focus)

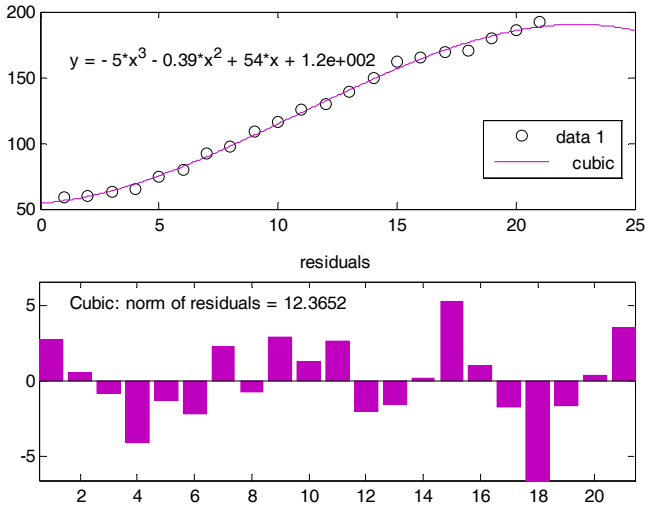
Full pattern period plot showing 15% intensity level



Intensity profile of stripe edge

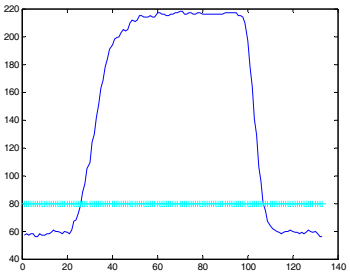


Cubic fit, plus residuals……..

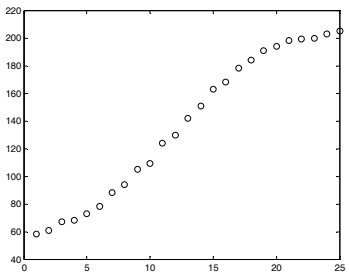


2) 110 cm from camera lens

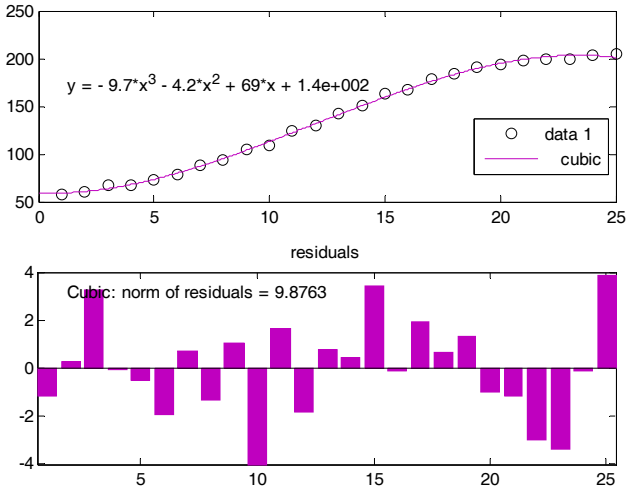
Pattern period plot @ 15%



Intensity profile of stripe edge

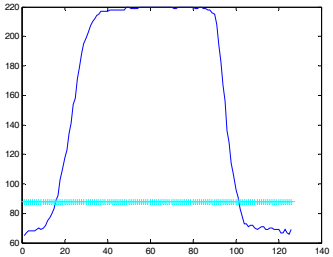


Cubic fit, plus residuals……..

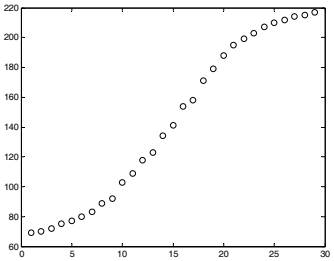


3) 105 cm from camera lens

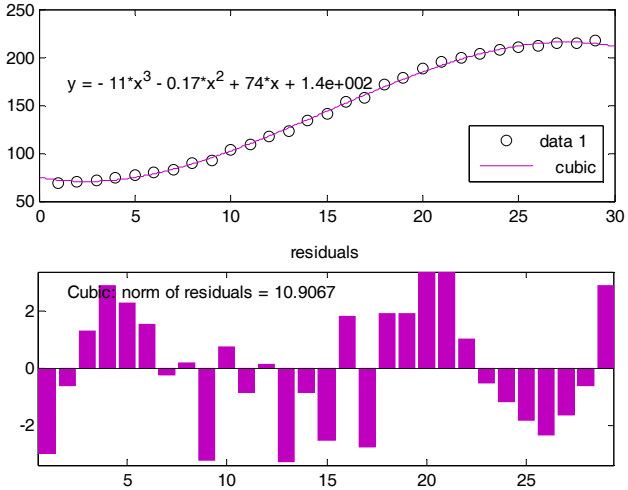
Pattern period plot @ 15%



Intensity profile of stripe edge

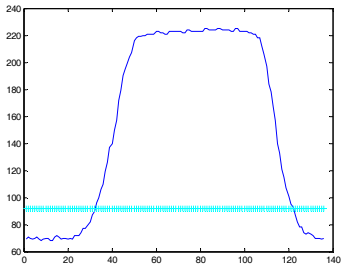


Cubic fit, plus residuals……..

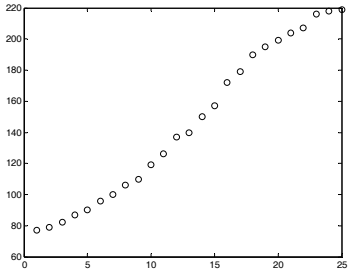


4) 100 cm from camera lens

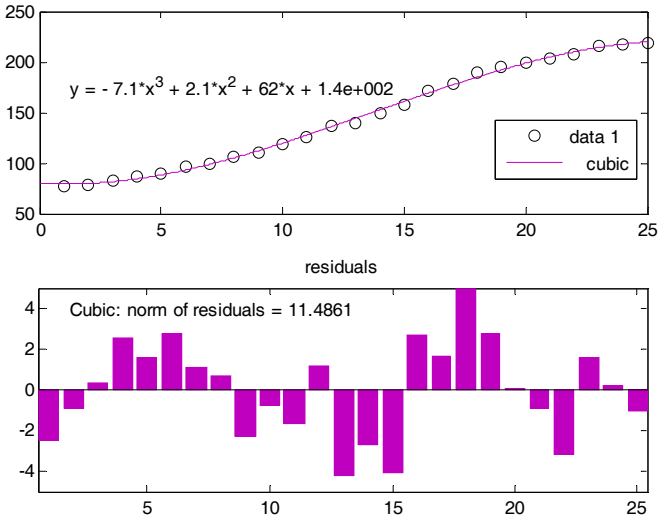
Pattern period plot @ 15%



Intensity profile of stripe edge

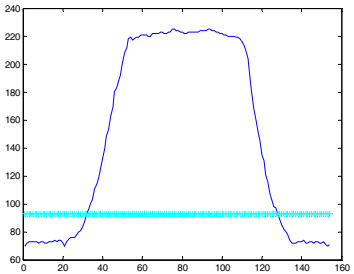


Cubic fit, plus residuals……..

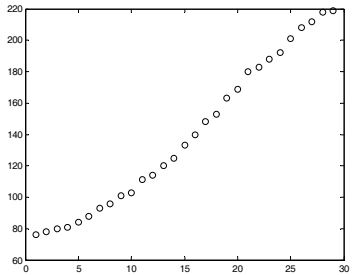


5) 95 cm from camera lens

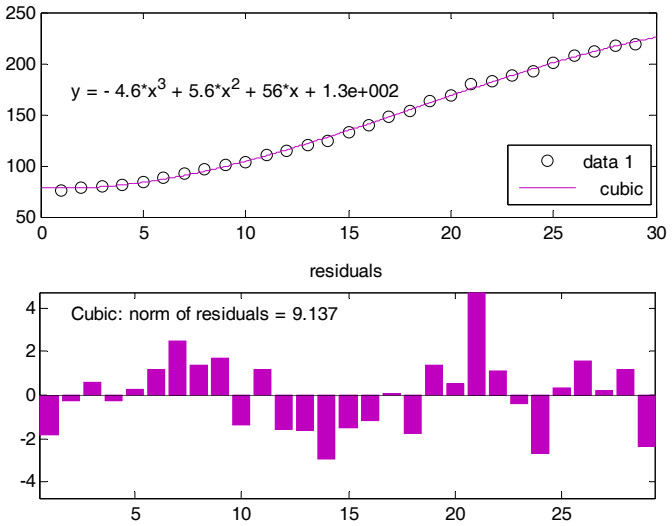
Pattern period plot @ 15%



Intensity profile of stripe edge

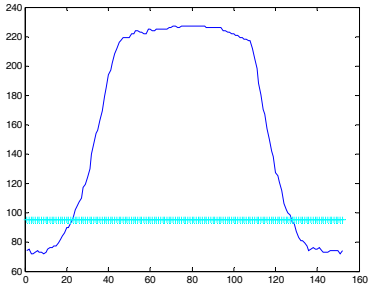


Cubic fit, plus residuals……..

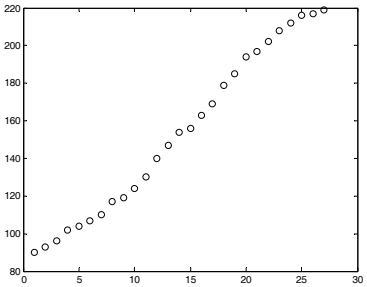


6) 90 cm from camera lens

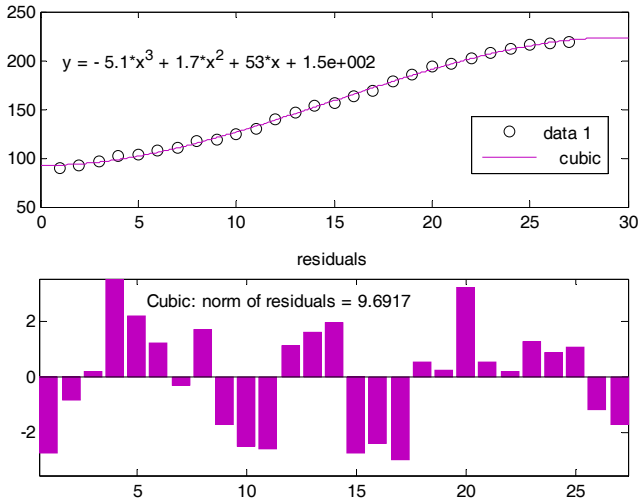
Pattern period plot @ 15%



Intensity profile of stripe edge

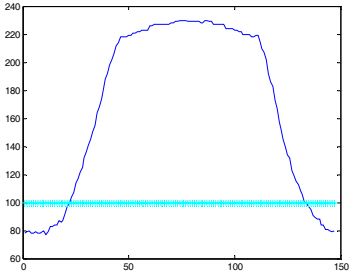


Cubic fit, plus residuals……..

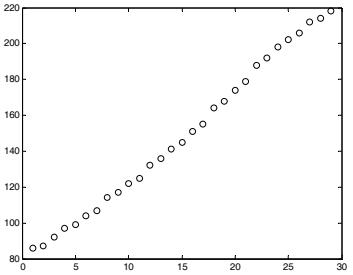


7) 85 cm from camera lens

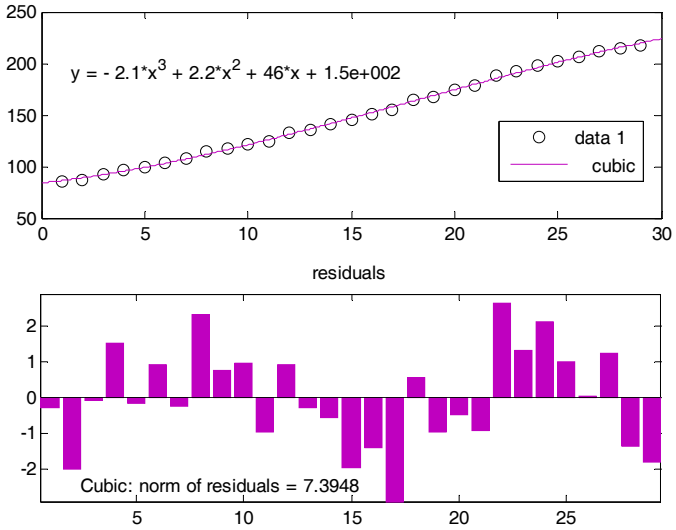
Pattern period plot @ 15%



Intensity profile of stripe edge

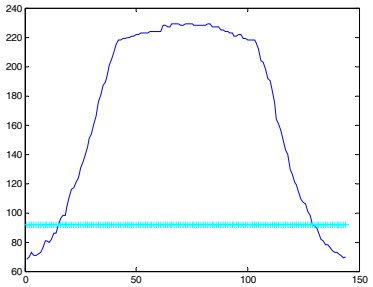


Cubic fit, plus residuals……..

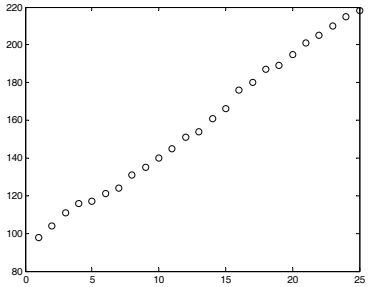


8) 80 cm from camera lens

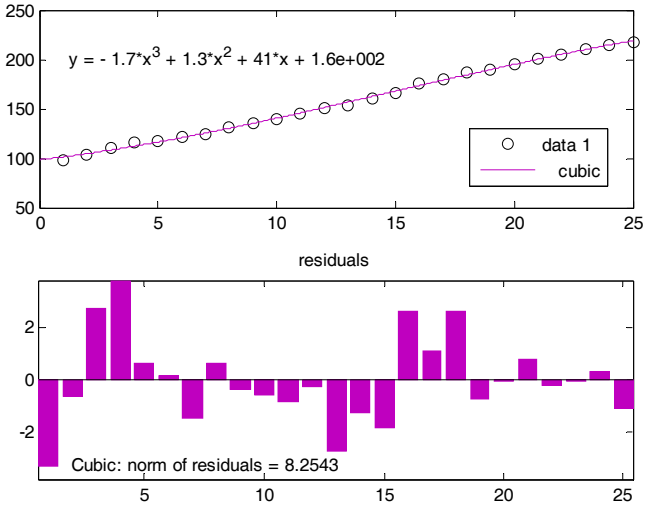
Pattern period plot @ 15%

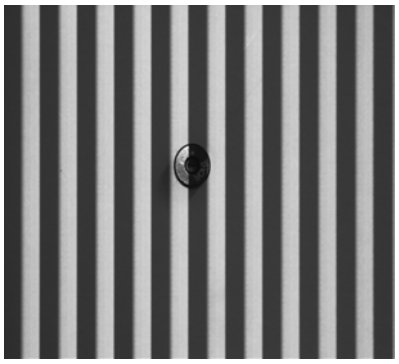


Intensity profile of stripe edge

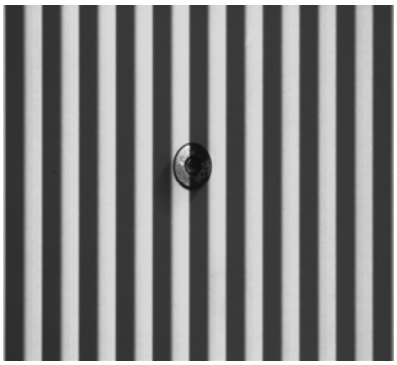


Cubic fit, plus residuals……..

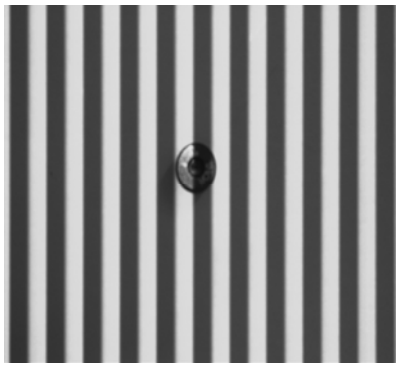




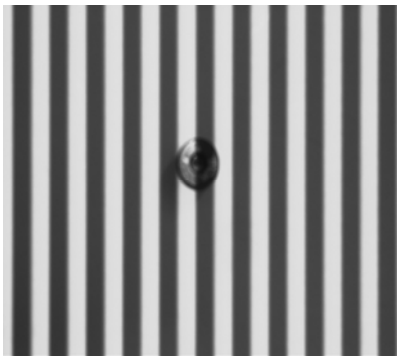
115 cm Depth (point of focus)



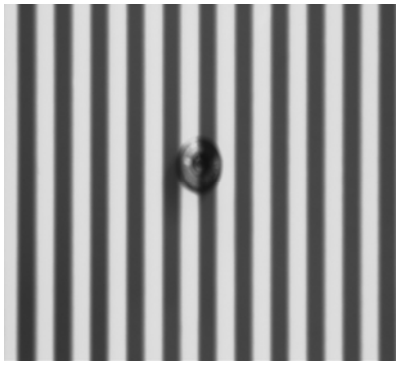
110 cm Depth



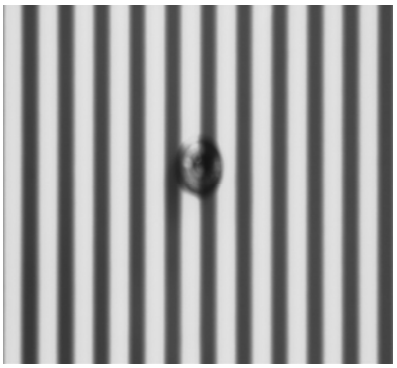
105 cm Depth



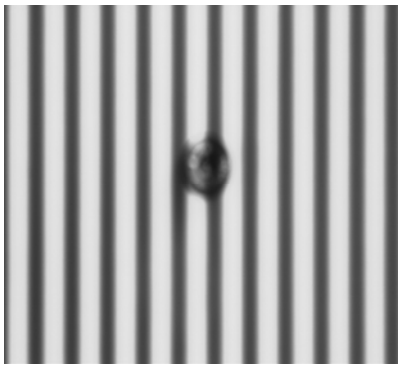
100 cm Depth



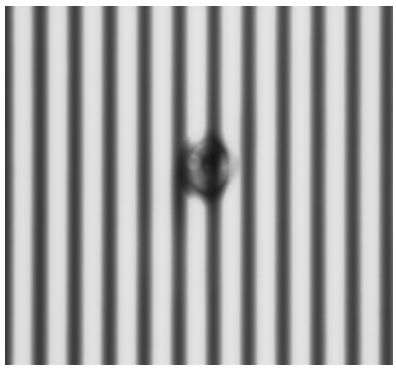
95 cm Depth



90 cm Depth



85 cm Depth

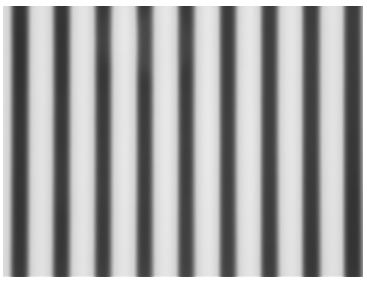


80 cm Depth

Images taken between depths of 115cm and 80cm show variation in white bar width as defocus varies. This causes a corresponding variation in the shape of the black-to-white edge profile, and allows this to be used as a depth cue.

Appendix B-4

The images shown below are produced from a static projected light pattern on a perpendicular flat plane of white background. Each image was then evaluated against a 2-Dimensional look-up table of 'depth standard' profiles to determine how accurately the range of edge profiles over all 1200 rows would be estimated to the known depth. For brevity, only every fifth image/estimate is shown in the depth range of 90 cm to 114 cm. In practice, estimates were taken over all 25 depths between 90 cm and 114 cm at intervals of 1 cm. The results are summarised in Table 5.1 and Figures 5.5 and 5.6



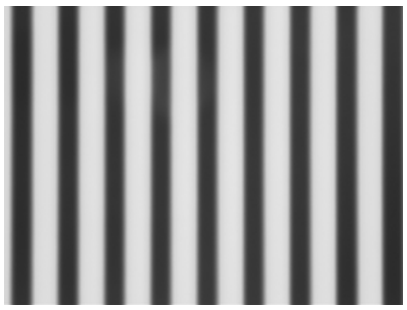
Depth = 90 cm

The mean of calculated depth values = 90.1183

The median of calculated depth values = 90

The standard deviation of calculated depth values = 0.34339

The percentage error in accuracy over the depth range = 0.38154%



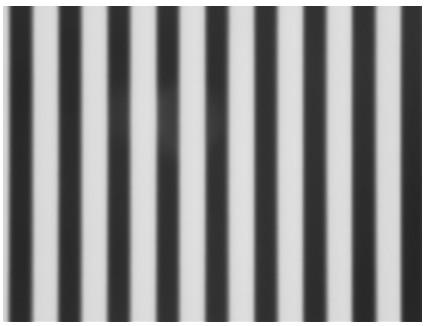
Depth = 95 cm

The mean of calculated depth values = 93.9254

The median of calculated depth values = 94

The standard deviation of calculated depth values = 1.1072

The percentage error in accuracy over the depth range = 1.1655%



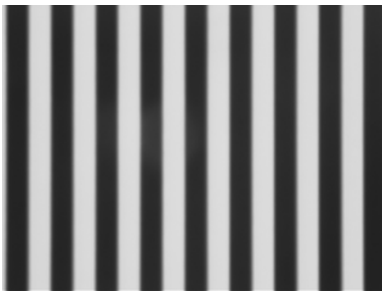
Depth = 100 cm

The mean of calculated depth values = 98.2895

The median of calculated depth values = 98

The standard deviation of calculated depth values = 1.4344

The percentage error in accuracy over the depth range = 1.4344%



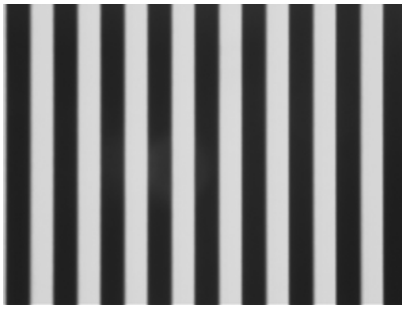
Depth = 105 cm

The mean of calculated depth values = 103.614

The median of calculated depth values = 103

The standard deviation of calculated depth values = 1.4343

The percentage error in accuracy over the depth range = 1.366%



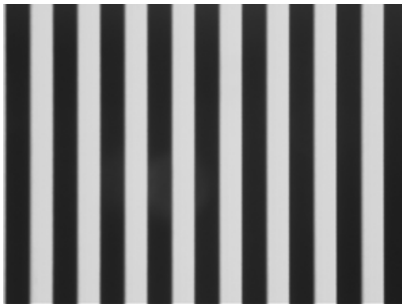
Depth = 110 cm

The mean of calculated depth values = 107.8825

The median of calculated depth values = 107

The standard deviation of calculated depth values = 1.5432

The percentage error in accuracy over the depth range = 1.4029%



Depth = 114 cm

The mean of calculated depth values = 110.0545

The median of calculated depth values = 110

The standard deviation of calculated depth values = 1.8985

The percentage error in accuracy over the depth range = 1.6653%

The results (plotted in Figure 5.5) show that the depth error was less than 0.5% at the nearest depth of 90 cm. The error gradually increased with depth, but remained at approximately 1.5% even at the furthest depths in the range. Depth estimation accuracies of less than 2% are very good for a first attempt with a single-image DfD system (see Table 3.4, Ch3). The standard deviation of the set of estimates per depth (see Figure 5.6) also indicated that the spread of estimate values was relatively low and was approximately proportional to the accuracy of the error.

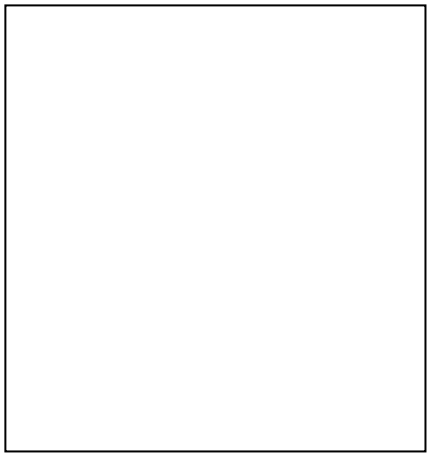
Appendix B-5

Results of Applying the 'imagecalc.m' Algorithm to Surfaces of Varying Absorption

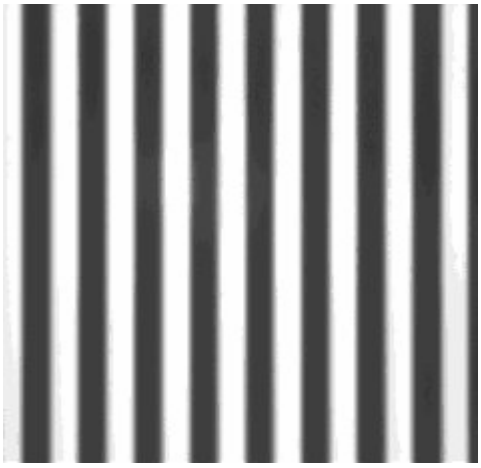
In order to determine how successfully the DfD algorithm using projected edge profiles would operate on surfaces of different reflectance, the following images were analysed. The structured light pattern was projected onto surfaces that had increasing levels of grey intensity. This meant that the intensity range between the 'white' and 'black' bars of the pattern was gradually decreasing as the background grey level increased. As each profile is normalised in its intensity range prior to analysis, it was conceivable that the normalised profile would retain enough of its shape due to defocus to enable it to be accurately matched to a 'standard' profile stored in the look-up table. All images were at a depth of 100 cm.

---

1. White background



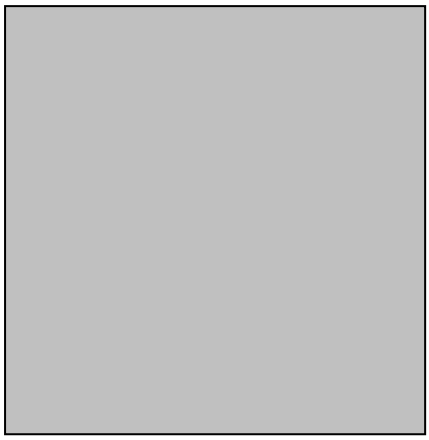
Background Screen

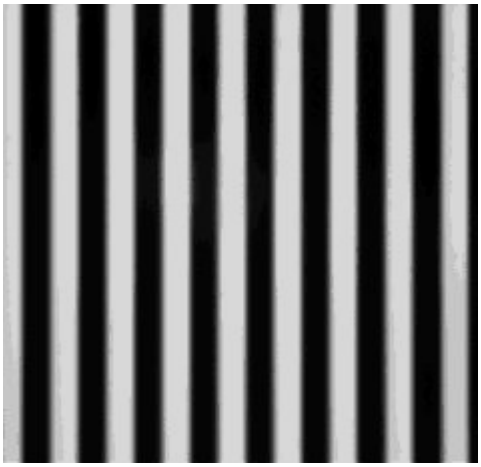


Captured Image of Projected Pattern

The mean of calculated depth values = 98.2895
The standard deviation of calculated depth values = 1.4344
The percentage error in accuracy over the depth range = 1.4344%

2. 25% Grey Background

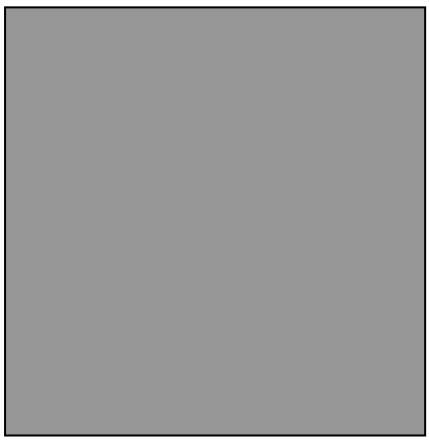


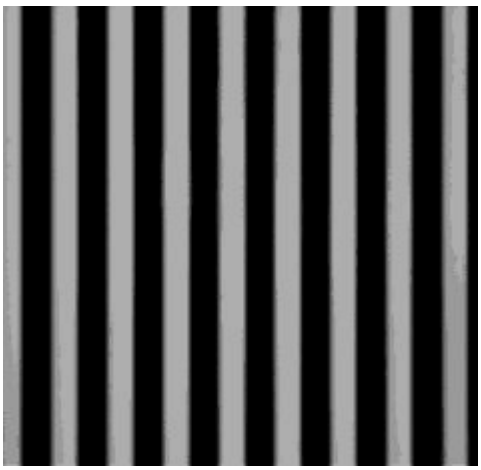


Background Screen Captured Image of Projected Pattern

The mean of calculated depth values = 102.5462
The standard deviation of calculated depth values = 1.5581
The percentage error in accuracy over the depth range = 1.6853%

3. 40% Grey Background

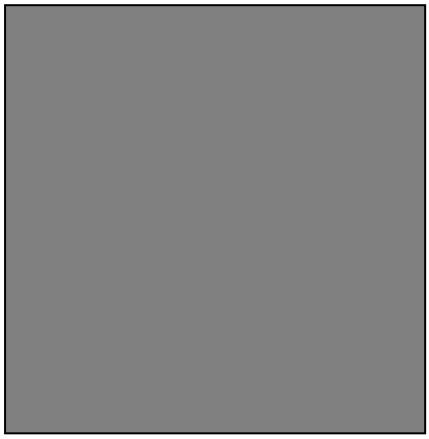


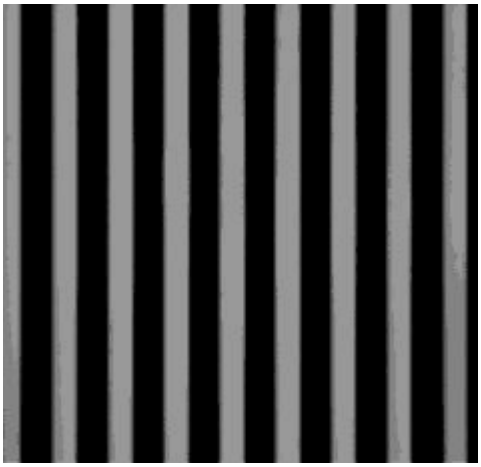


Background Screen Captured Image of Projected Pattern

The mean of calculated depth values = 101.7284
The standard deviation of calculated depth values = 1.6829
The percentage error in accuracy over the depth range = 1.7256%

4. 50% Grey Background

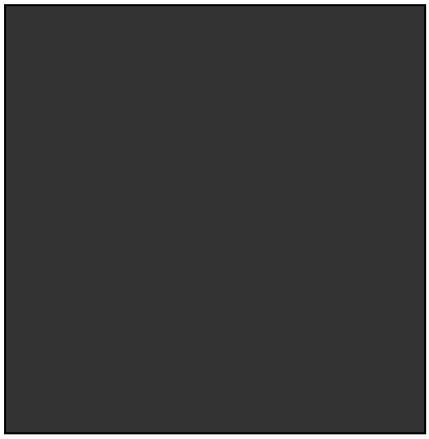


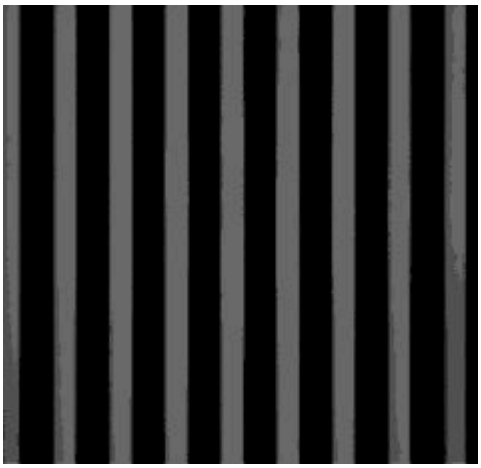


Background Screen Captured Image of Projected Pattern

The mean of calculated depth values = 100.3085
The standard deviation of calculated depth values = 1.7927
The percentage error in accuracy over the depth range = 1.7619%

5. 80% Grey Background

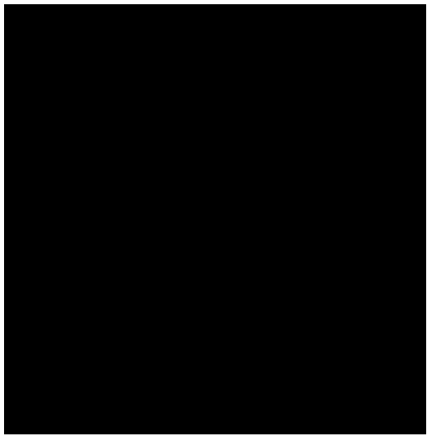


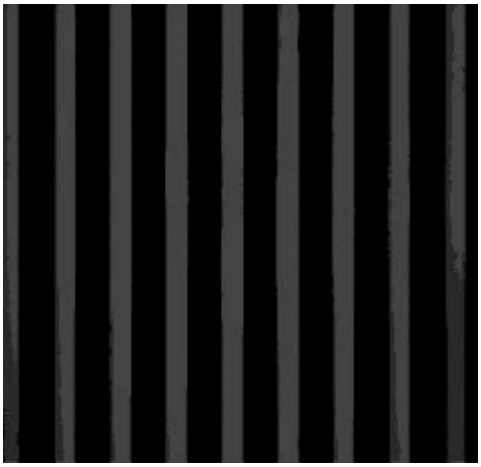


Background Screen Captured Image of Projected Pattern

The mean of calculated depth values = 104.7382
The standard deviation of calculated depth values = 2.1599
The percentage error in accuracy over the depth range = 3.4638%

6. Black Background





Background Screen Captured Image of Projected Pattern

The mean of calculated depth values = 107.6209
The standard deviation of calculated depth values = 2.5743
The percentage error in accuracy over the depth range = 5.1348%

The results illustrated above show that the algorithm continues to operate reasonably successfully up to and including the image taken from a 50% grey background. However, it should be noted that – although the mean depth estimate values for images with backgrounds up to 50% grey level remained within approximately 2.5% of the actual depth – the standard deviation of all estimates per image increased monotonically as the background became darker. A plot of the standard deviation and % error versus background absorption is given in Figure 5.7.

Appendix B-6

Effect of Surface Angle on the Accuracy of DfD Estimation

A white planar surface was set up at a distance of 96cm from the camera lens. The surface was positioned on the rotational axis of a shaft mounted in a vertical holder. The field of view of the camera was centred exactly on the rotational axis of the object, and the projected (vertical bar) pattern was positioned such that a rising edge was also directly located on the centre of rotation. An image was captured with the surface being first at 0^o to the optical axis, then 20^o, 30^o, 40^o, 50^o, and 60^o respectively. The results – shown below – show that the percentage error for surface slopes of up to 60^o of rotation from the camera axis are within 1½ % accuracy.

Load first image for analysis? Enter 1 to continue or 0 to end > 1
Image 096cm_0rot.bmp has been analysed % 0 degree rotation
Average depth profile for 096cm_0rot.bmp = 96 cm
Error in average recorded depth is 0 %

Load another image? Enter 1 to continue or 0 to end > 1
Image 096cm_10rot.bmp has been analysed % 10 degree rotation
Average depth profile for 096cm_10rot.bmp = 94.9438 cm
Error in average recorded depth is 1.1002 %

Load another image? Enter 1 to continue or 0 to end > 1
Image 096cm_20rot.bmp has been analysed % 20 degree rotation
Average depth profile for 096cm_20rot.bmp = 94.9665 cm
Error in average recorded depth is 1.0765 %

Load another image? Enter 1 to continue or 0 to end > 1
Image 096cm_30rot.bmp has been analysed % 30 degree rotation
Average depth profile for 096cm_30rot.bmp = 94.9295 cm
Error in average recorded depth is 1.1151 %

Load another image? Enter 1 to continue or 0 to end > 1
Image 096cm_40rot.bmp has been analysed % 40 degree rotation
Average depth profile for 096cm_40rot.bmp = 94.9403 cm
Error in average recorded depth is 1.1039 %

Load another image? Enter 1 to continue or 0 to end > 1
Image 096cm_50rot.bmp has been analysed % 50 degree rotation
Average depth profile for 096cm_50rot.bmp = 94.9379 cm
Error in average recorded depth is 1.1064 %

Load another image? Enter 1 to continue or 0 to end > 1
Image 096cm_60rot.bmp has been analysed % 60 degree rotation
Average depth profile for 096cm_60rot.bmp = 94.6117 cm
Error in average recorded depth is 1.4461 %

Load another image? Enter 1 to continue or 0 to end > 1
Image 096cm_70rot.bmp has been analysed % 70 degree rotation
Average depth profile for 096cm_70rot.bmp = 90.3864 cm
Error in average recorded depth is 5.8475 %

Load another image? Enter 1 to continue or 0 to end > 0

Appendix B-7

This experiment was carried out in order to determine whether improvements could be made to the accuracy of depth estimates by the process of averaging time-lapsed images of the same depth. Sixteen images of the structured light pattern were taken at time intervals of approximately 15 seconds (time to capture, label and save the image). The images were then averaged accumulatively, and were analysed via the DfD algorithm until all 16 images had been included in the averaged image.

Averaging 16 Images at 110cm Planar Depth

Image analysed is 110cm01.bmp
The known depth is 110 cm
The minimum estimated depth value = 105 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.5181 cm
The standard deviation of calculated depth values = 2.553
The ESTIMATED DEPTH VALUE = 111 cm
The estimated residuals = -5.5181 cm to 3.4819 cm
The ERROR = 0.47096%

Image analysed is 110cm02.bmp
The known depth is 110 cm
The minimum estimated depth value = 104 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.359 cm
The standard deviation of calculated depth values = 2.4565
The ESTIMATED DEPTH VALUE = 110 cm
The estimated residuals = -6.359 cm to 3.641 cm
The ERROR = 0.32639%

Image analysed is 110cm03.bmp
The known depth is 110 cm
The minimum estimated depth value = 105 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.8292 cm
The standard deviation of calculated depth values = 2.4675
The ESTIMATED DEPTH VALUE = 111 cm
The estimated residuals = -5.8292 cm to 3.1708 cm
The ERROR = 0.75379%

Image analysed is 110cm04.bmp
The known depth is 110 cm
The minimum estimated depth value = 104 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.3678 cm
The standard deviation of calculated depth values = 2.4667
The ESTIMATED DEPTH VALUE = 110 cm
The estimated residuals = -6.3678 cm to 3.6322 cm
The ERROR = 0.33434%

Image analysed is 110cm05.bmp
The known depth is 110 cm
The minimum estimated depth value = 104 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.3676 cm
The standard deviation of calculated depth values = 2.4383
The ESTIMATED DEPTH VALUE = 110 cm
The estimated residuals = -6.3676 cm to 3.6324 cm
The ERROR = 0.33422%

Image analysed is 110cm06.bmp
The known depth is 110 cm
The minimum estimated depth value = 105 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.4139 cm
The standard deviation of calculated depth values = 2.4612
The ESTIMATED DEPTH VALUE = 110 cm
The estimated residuals = -5.4139 cm to 3.5861 cm
The ERROR = 0.37626%

Image analysed is 110cm07.bmp
The known depth is 110 cm
The minimum estimated depth value = 105 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.5846 cm
The standard deviation of calculated depth values = 2.464
The ESTIMATED DEPTH VALUE = 111 cm
The estimated residuals = -5.5846 cm to 3.4154 cm
The ERROR = 0.53144%

Image analysed is 110cm08.bmp
The known depth is 110 cm
The minimum estimated depth value = 104 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.4239 cm
The standard deviation of calculated depth values = 2.5051
The ESTIMATED DEPTH VALUE = 110 cm
The estimated residuals = -6.4239 cm to 3.5761 cm
The ERROR = 0.38535%

Image analysed is 110cm09.bmp
The known depth is 110 cm
The minimum estimated depth value = 105 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.3549 cm
The standard deviation of calculated depth values = 2.4686
The ESTIMATED DEPTH VALUE = 110 cm
The estimated residuals = -5.3549 cm to 3.6451 cm
The ERROR = 0.3226%

Image analysed is 110cm10.bmp
The known depth is 110 cm
The minimum estimated depth value = 104 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.4121 cm
The standard deviation of calculated depth values = 2.4695
The ESTIMATED DEPTH VALUE = 110 cm
The estimated residuals = -6.4121 cm to 3.5879 cm
The ERROR = 0.37462%

Image analysed is 110cm11.bmp
The known depth is 110 cm
The minimum estimated depth value = 104 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.4464 cm
The standard deviation of calculated depth values = 2.4841
The ESTIMATED DEPTH VALUE = 110 cm
The estimated residuals = -6.4464 cm to 3.5536 cm
The ERROR = 0.40581%

Image analysed is 110cm12.bmp
The known depth is 110 cm
The minimum estimated depth value = 104 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.384 cm
The standard deviation of calculated depth values = 2.4672
The ESTIMATED DEPTH VALUE = 110 cm
The estimated residuals = -6.384 cm to 3.616 cm
The ERROR = 0.34912%

Image analysed is 110cm13.bmp
The known depth is 110 cm
The minimum estimated depth value = 105 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.5331 cm
The standard deviation of calculated depth values = 2.4996
The ESTIMATED DEPTH VALUE = 111 cm
The estimated residuals = -5.5331 cm to 3.4669 cm
The ERROR = 0.4846%

Image analysed is 110cm14.bmp
The known depth is 110 cm
The minimum estimated depth value = 104 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.3764 cm
The standard deviation of calculated depth values = 2.4684
The ESTIMATED DEPTH VALUE = 110 cm
The estimated residuals = -6.3764 cm to 3.6236 cm
The ERROR = 0.34217%

Image analysed is 110cm15.bmp
The known depth is 110 cm
The minimum estimated depth value = 105 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.4069 cm
The standard deviation of calculated depth values = 2.4664
The ESTIMATED DEPTH VALUE = 110 cm
The estimated residuals = -5.4069 cm to 3.5931 cm
The ERROR = 0.36995%

Image analysed is 110cm16.bmp
The known depth is 110 cm
The minimum estimated depth value = 104 cm
The maximum estimated depth value = 114 cm
The mean of calculated depth values = 110.3378 cm
The standard deviation of calculated depth values = 2.4405
The ESTIMATED DEPTH VALUE = 110 cm
The estimated residuals = -6.3378 cm to 3.6622 cm
The ERROR = 0.30707%

Conclusion

The error results, plotted in Figure 5.11, show no particular trend of improvement in estimated depth error as a result of time-lapsed averaging, and do not suggest that there would be any benefit in including this process in the algorithm. Reasons for this are discussed in section 5.5 of chapter 5.

Appendix B-8

Mean of Depth Estimates from Normalised Profiles Averaged over 16 Images

Note: The accumulator vector is storing the depth estimate for each averaged profile or 'column' as the averaging progresses from image 1 through to image 16. Each of the values shown represents the sum of the profile plot between the normalised 'x' values in the range 30 to 50 pixels. Lower values represent greater depths.

Image1	**20.6467**	**21.3800**	**22.9233**	**22.3650**	**21.3092**	**20.4842**
Acc Ave:	20.6467	21.3800	22.9233	22.3650	21.3092	20.4842
Image2	**20.7092**	**21.3908**	**22.7500**	**22.1725**	**21.8833**	**20.2483**
Acc Ave:	20.6779	21.3854	22.8367	22.2688	21.0963	20.3663
Image3	**20.0950**	**21.7717**	**22.1517**	**22.5525**	**21.6758**	**20.7283**
Acc Ave:	20.8865	21.5785	22.9942	22.4106	21.3860	20.5473
Image4	**20.7483**	**21.3800**	**22.7150**	**22.2775**	**21.8000**	**20.2858**
Acc Ave:	20.8174	21.4793	22.8546	22.3441	21.0930	20.4166
Image5	**20.7658**	**21.4058**	**22.7575**	**22.1617**	**21.7700**	**20.3450**
Acc Ave:	20.7916	21.4426	22.8060	22.2529	21.9315	20.3808
Image5	**20.7708**	**21.4200**	**22.8050**	**22.1858**	**21.9233**	**20.3783**
Acc Ave:	20.7812	21.4313	22.8055	22.2193	21.9274	20.3796
Image6	**20.8967**	**21.5667**	**22.9142**	**22.4350**	**21.2075**	**20.4875**
Acc Ave:	20.8389	21.4990	22.8598	22.3272	21.0675	20.4335
Image7	**20.6967**	**21.3358**	**22.8183**	**22.3725**	**21.9350**	**20.3850**
Acc Ave:	20.8389	21.4990	22.8598	22.3272	21.0675	20.4335
Image8	**20.6967**	**21.3358**	**22.8183**	**22.3725**	**21.9350**	**20.3850**
Acc Ave:	20.7678	21.4174	22.8391	22.3498	21.0012	20.4093
Image9	**20.6783**	**21.4017**	**22.7058**	**22.1400**	**21.8142**	**20.3892**
Acc Ave:	20.7231	21.4095	22.7725	22.2449	21.9077	20.3992
Image10	**20.7267**	**21.3908**	**22.7525**	**22.2858**	**21.9775**	**20.3392**
Acc Ave:	20.7249	21.4002	22.7625	22.2654	21.9426	20.3692
Image11	**20.7525**	**21.4308**	**22.8008**	**22.2525**	**21.9992**	**20.4425**
Acc Ave:	20.7387	21.4155	22.7817	22.2589	21.9709	20.4058
Image12	**20.7058**	**21.3958**	**22.7458**	**22.2808**	**21.8658**	**20.3100**
Acc Ave:	20.7223	21.4057	22.7637	22.2699	21.9184	20.3579
Image13	**20.7767**	**21.5008**	**22.8525**	**22.4067**	**21.1200**	**20.5417**
Acc Ave:	20.7495	21.4533	22.8081	22.3383	21.0192	20.4498
Image14	**20.6542**	**21.3892**	**22.7475**	**22.2667**	**21.8867**	**20.3142**
Acc Ave:	20.7018	21.4212	22.7778	22.3025	21.9529	20.3820
Image15	**20.6700**	**21.4425**	**22.8275**	**22.2433**	**21.9100**	**20.3483**
Acc Ave:	20.6859	21.4319	22.8027	22.2729	22.9315	20.3652
Image16	**20.6492**	**21.4108**	**22.6783**	**22.3033**	**21.7617**	**20.2233**
Acc Ave:	20.6675	21.4213	22.7405	22.2881	21.8466	20.2942

Appendix B-9

Evaluating the cause of variation in depth estimate across image rows

Figure B1.1 shows the plot of the profile of the first rising edge detected on the row of an image of the projected structured light pattern. The profile has already been normalised in '*y*' (greyscale intensity), and hence the figure shows the normalised value of the extracted look-up summation that will be used to find the closest match of the profile with a 'standard' profile from the DfD look-up table.

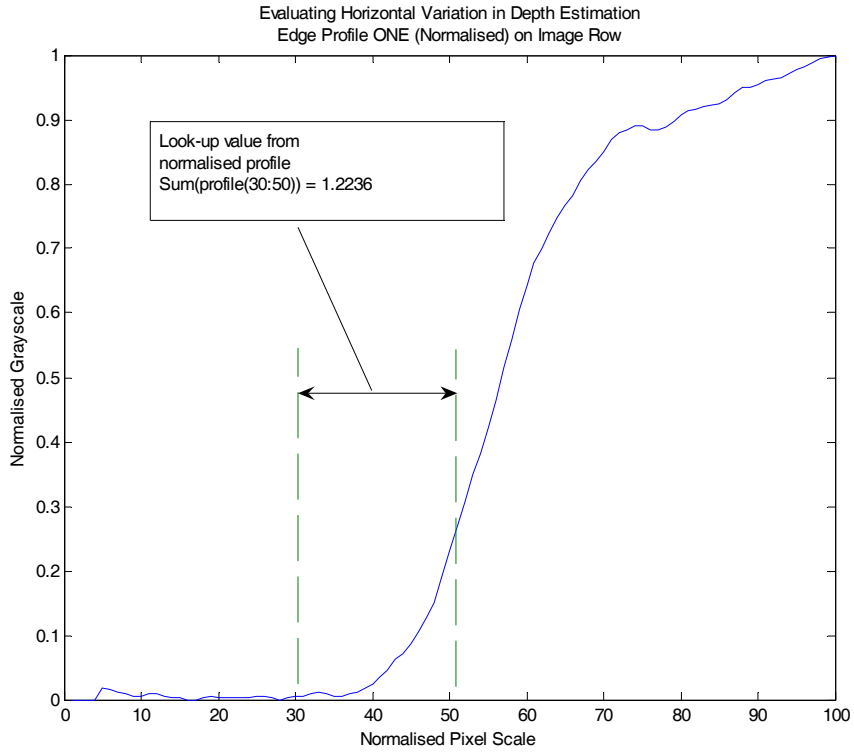


Figure B1.1 Edge Profile Plot showing extracted look-up value

The plot of normalised look-up value summations for all six edge profiles can be seen in Figure B1.2 below.

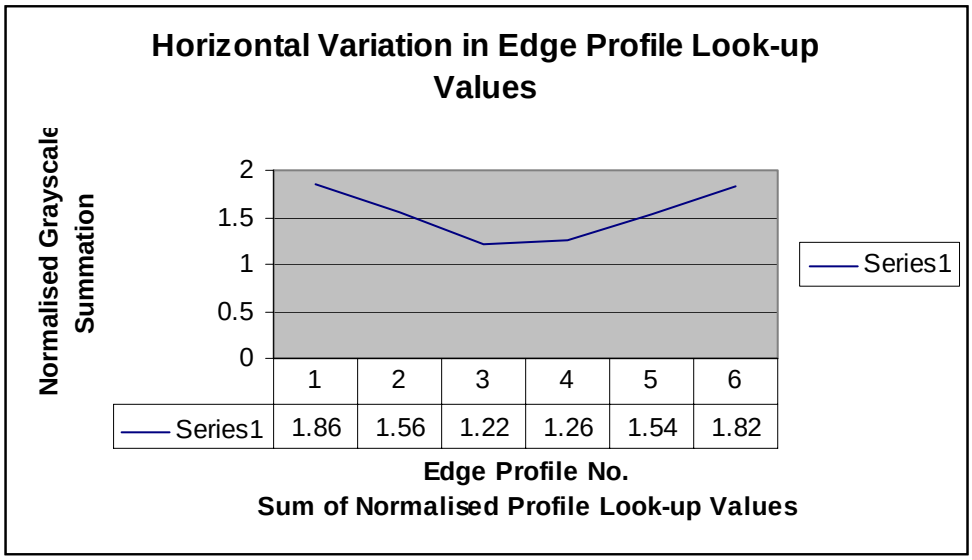


Figure B1.2 Plot of six edge profile look-up values across image row

Appendix B-10

Evaluating the Improvement in Horizontal Variability of Depth Estimation using a 3-Dimensional Look-up Table

The following data were collected by applying the 3-D look-up table algorithm to a set of 20 images of a spatially shifted structured light pattern that was projected onto a planar screen at a distance of 100 cm from the camera lens. This distance corresponds to a look-up table index of 11.
The 3-D look-up table system extracts 200 edge profiles per image row for each of 1200 rows, and the depth estimate (in terms of look-up table index) for each profile is averaged with the other 1199 estimates produced by edge profiles from the same horizontal position or 'column' on each row.
The table below records the average depth estimate in terms of look-up table index for each of the 200 profile columns in the image.

Col No	Index	Col No	Index	Col No	Index	Col No	Index	Col No	Index
1	10.9398	41	10.9434	81	11.0917	121	10.9374	161	11.0941
2	11.0349	42	11.0304	82	11.036	122	11.0983	162	10.9974
3	11.0854	43	10.9106	83	10.911	123	11.0424	163	11.0635
4	10.9688	44	10.9459	84	11.02	124	11.0743	164	11.0283
5	11.0189	45	11.0335	85	10.9786	125	10.9959	165	10.9613
6	11.0231	46	10.9622	86	10.9431	126	10.9992	166	11.0322
7	10.9007	47	10.9613	87	10.9365	127	10.9575	167	10.9716
8	11.0964	48	11.0441	88	10.9154	128	10.9122	168	11.0876
9	11.0799	49	11.0909	89	10.9015	129	10.9525	169	10.9975
10	11.0386	50	10.9262	90	11.0578	130	10.9373	170	10.9182
11	10.9879	51	10.9137	91	10.9036	131	11.0834	171	11.0348
12	11.0402	52	10.925	92	11.0756	132	10.9247	172	11.003
13	11.0219	53	10.9332	93	10.9705	133	10.9027	173	10.9443
14	10.96	54	11.0823	94	11.0444	134	10.9739	174	11.045
15	11.0712	55	10.9273	95	11.0937	135	11.0397	175	10.9136
16	10.9224	56	11.0234	96	10.9311	136	11.0779	176	11.0928
17	10.9583	57	10.9538	97	10.9326	137	11.0188	177	10.9415
18	10.9195	58	10.9441	98	10.9627	138	10.9313	178	10.9322
19	10.9795	59	11.0426	99	10.9059	139	10.9633	179	11.0276
20	10.9667	60	11.0098	100	10.9715	140	10.9467	180	10.9
21	11.0888	61	11.0883	101	10.9054	141	10.9017	181	10.9671
22	11.0677	62	10.966	102	11.0587	142	10.9794	182	10.955
23	10.9517	63	11.0409	103	11.0998	143	11.03	183	10.9089
24	10.9086	64	11.0887	104	10.922	144	10.917	184	10.9188
25	10.9012	65	11.0163	105	11.0245	145	11.0538	185	10.982
26	11.0149	66	11.076	106	10.9265	146	11.0939	186	11.0634
27	11.0488	67	11.0499	107	10.962	147	11.043	187	11.0741
28	11.0614	68	10.9759	108	10.927	148	11.0564	188	10.9045
29	11.0275	69	11.0451	109	10.9447	149	10.9475	189	11.0454
30	10.9503	70	10.9326	110	10.9793	150	10.9391	190	11.0696
31	10.9289	71	11.0912	111	10.927	151	10.9526	191	11.0457
32	11.0303	72	10.9392	112	10.9482	152	11.0428	192	11.091
33	11.0892	73	11.0552	113	11.0855	153	11.0955	193	11.0313
34	11.0632	74	11.0227	114	10.9782	154	11.0274	194	11.0485
35	11.086	75	10.9325	115	11.0023	155	11.0092	195	10.969
36	10.962	76	10.9062	116	10.9186	156	11.0696	196	11.0768
37	10.9538	77	10.9577	117	10.9043	157	11.0604	197	10.9694
38	11.0073	78	11.0942	118	10.9319	158	11.0337	198	10.9119
39	10.9327	79	11.0901	119	11.0689	159	11.0342	199	11.0437
40	10.9422	80	10.9456	120	11.0758	160	11.0641	200	11.0916

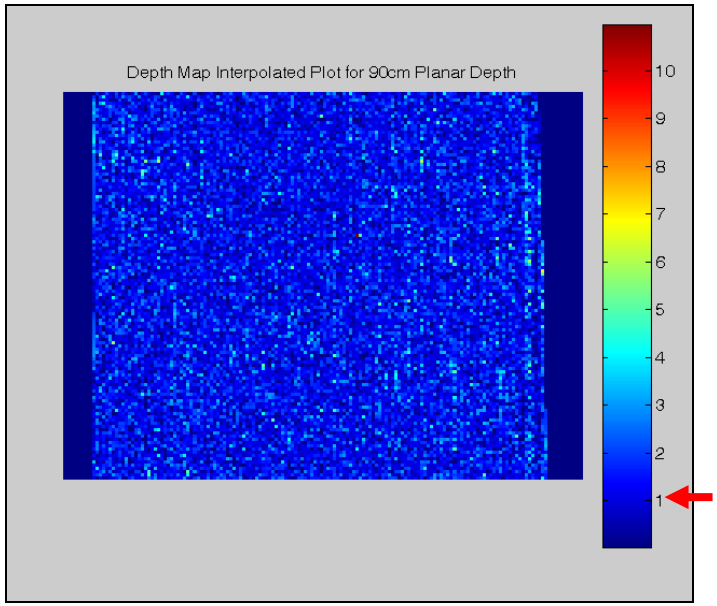
Appendix B-11 Evaluation of 4D LUT

Evaluating Planar Depths over 10 depths of 1 cm interval

(Red arrows show look-up table index at that depth)

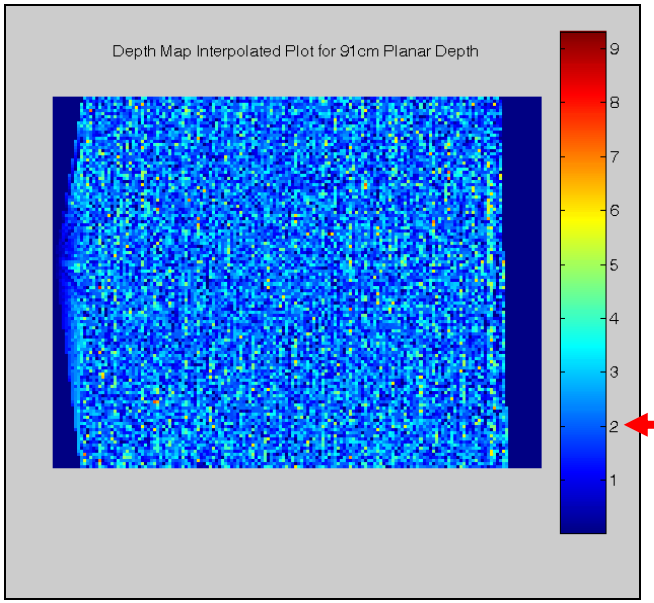
90cm

The known depth is 90 cm
The mean of calculated depth values = 90.337 cm
The standard deviation of calculated depth values = 1.04
R.M.S. Error = 1.0932
Percentage R.M.S. Error = 1.2147%
Calculated Depth Estimate Error = 0.37449%



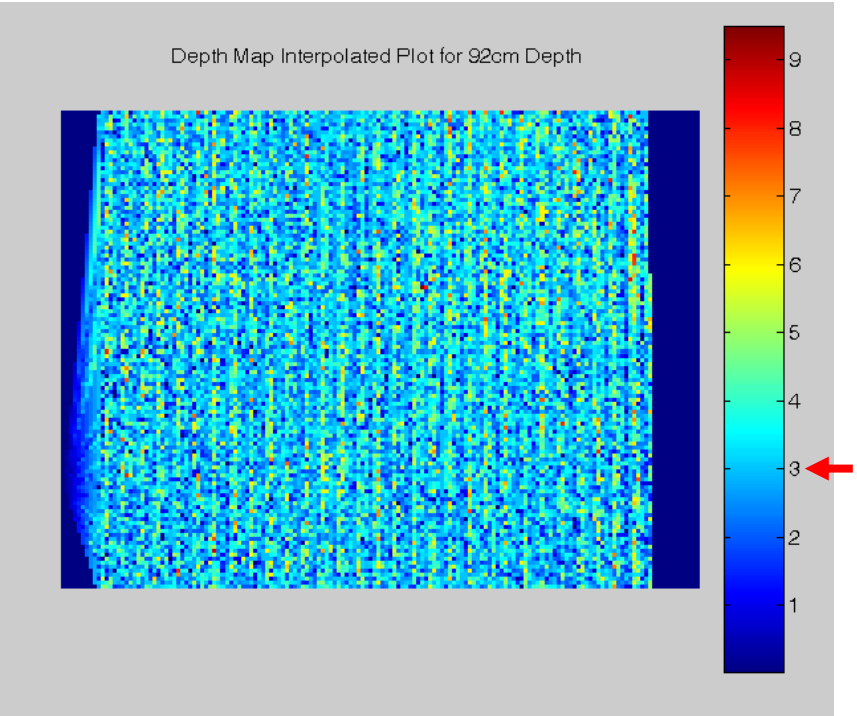
91cm

The known depth is 91 cm
The mean of calculated depth values = 91.208 cm
The standard deviation of calculated depth values = 1.3516
R.M.S. Error = 1.3675
Percentage R.M.S. Error = 1.5028%
Calculated Depth Estimate Error = 0.22861%



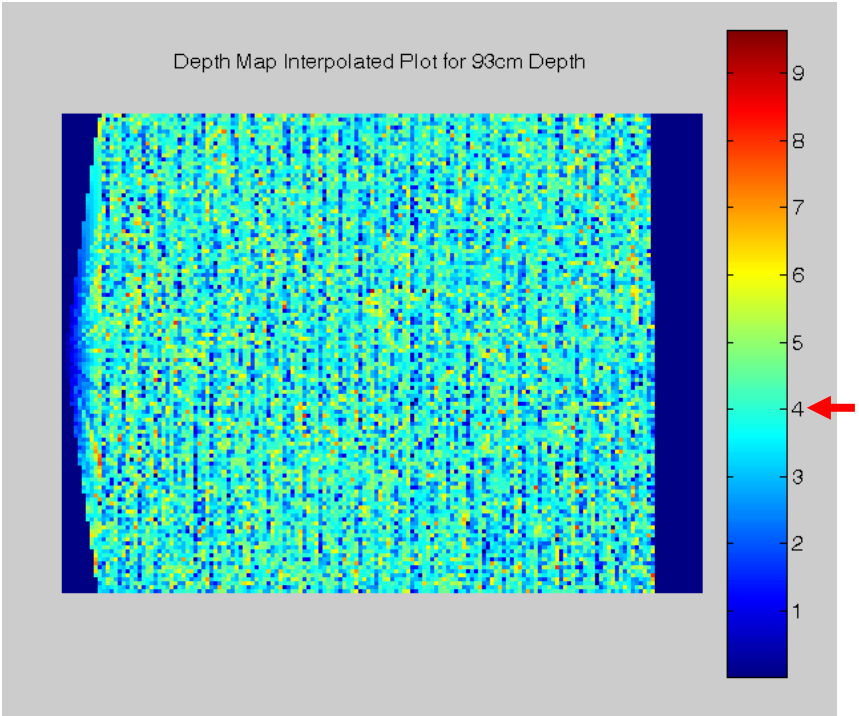
92cm

The known depth is 92 cm
The mean of calculated depth values = 92.2621 cm
The standard deviation of calculated depth values = 1.5024
R.M.S. Error = 1.5251
Percentage R.M.S. Error = 1.6577%
Calculated Depth Estimate Error = 0.28492%



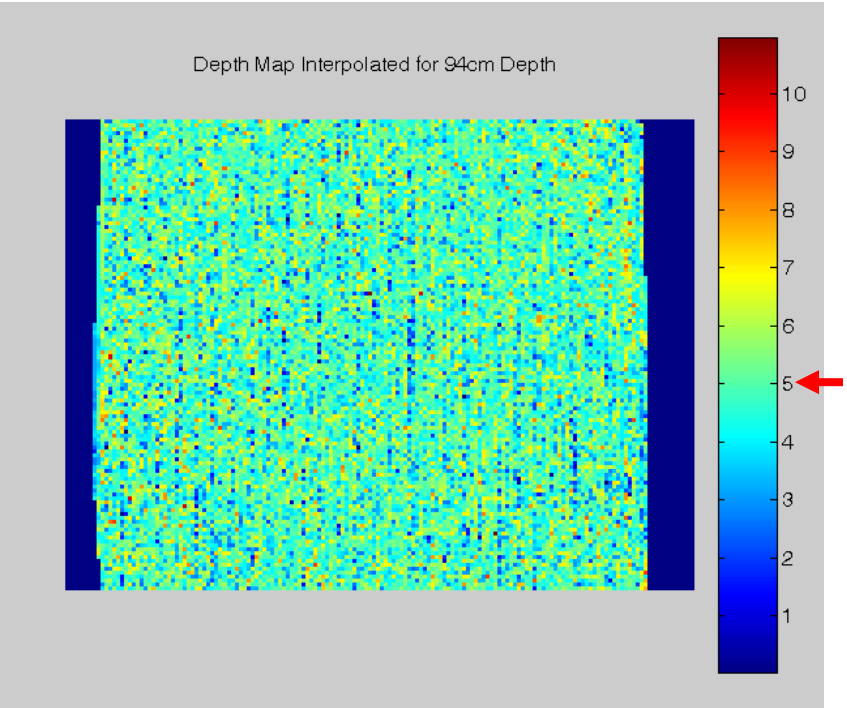
93cm

The known depth is 93 cm
The mean of calculated depth values = 92.8476 cm
The standard deviation of calculated depth values = 1.4868
R.M.S. Error = 1.4946
Percentage R.M.S. Error = 1.6071%
Calculated Depth Estimate Error = -0.16383%



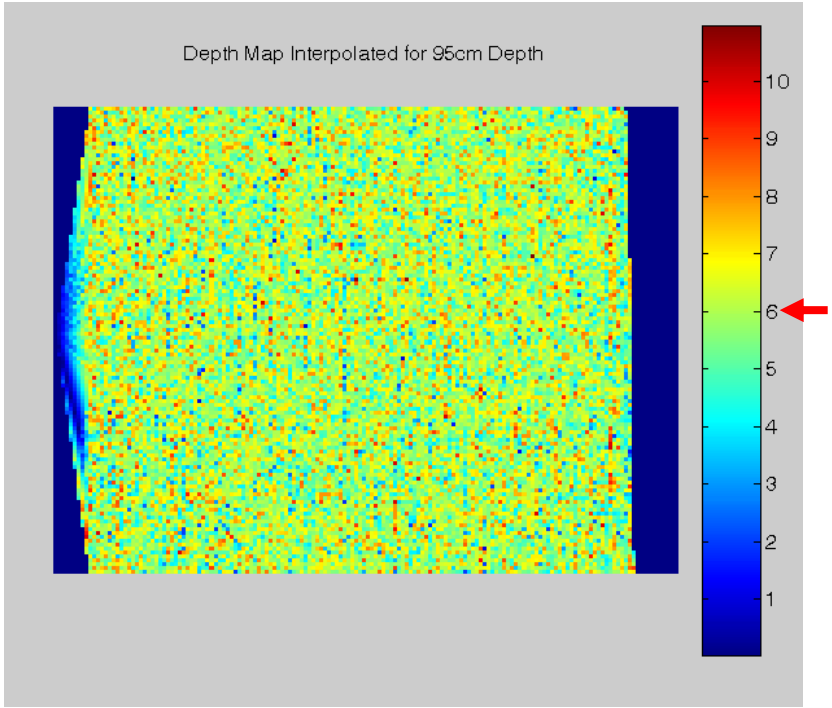
94cm

The known depth is 94 cm
The mean of calculated depth values = 93.9027 cm
The standard deviation of calculated depth values = 1.4829
R.M.S. Error = 1.4861
Percentage R.M.S. Error = 1.5809%
Calculated Depth Estimate Error = -0.10354%



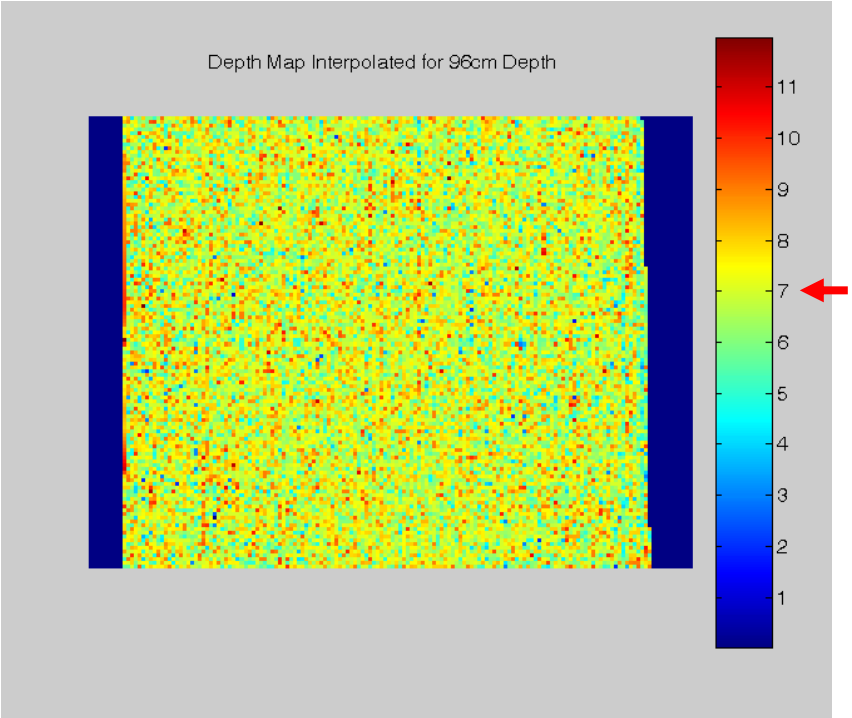
95cm

The known depth is 95 cm
The mean of calculated depth values = 94.9712 cm
The standard deviation of calculated depth values = 1.4871
R.M.S. Error = 1.4874
Percentage R.M.S. Error = 1.5656%
Calculated Depth Estimate Error = -0.03032%



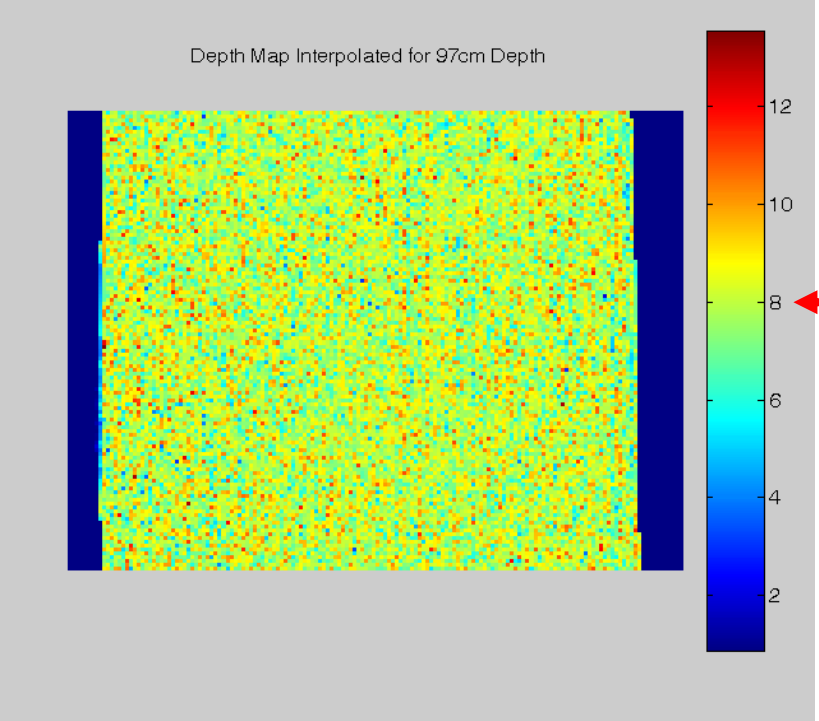
96cm

The known depth is 96 cm
The mean of calculated depth values = 96.0003 cm
The standard deviation of calculated depth values = 1.461
R.M.S. Error = 1.461
Percentage R.M.S. Error = 1.5218%
Calculated Depth Estimate Error = 0.00029948%



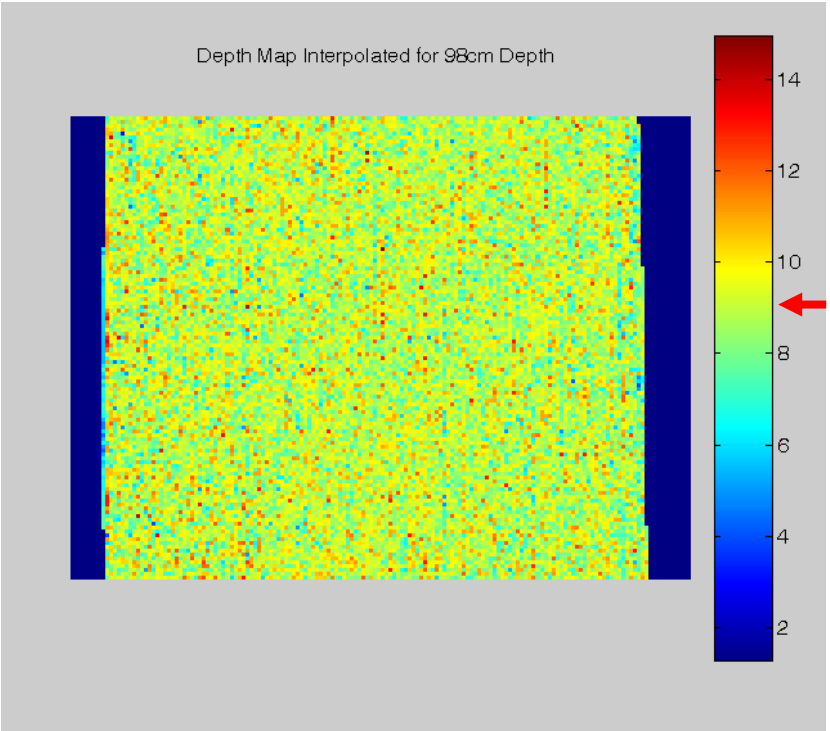
97cm

The known depth is 97 cm
The mean of calculated depth values = 97.0052 cm
The standard deviation of calculated depth values = 1.4716
R.M.S. Error = 1.4716
Percentage R.M.S. Error = 1.5171%
Calculated Depth Estimate Error = 0.0053651%



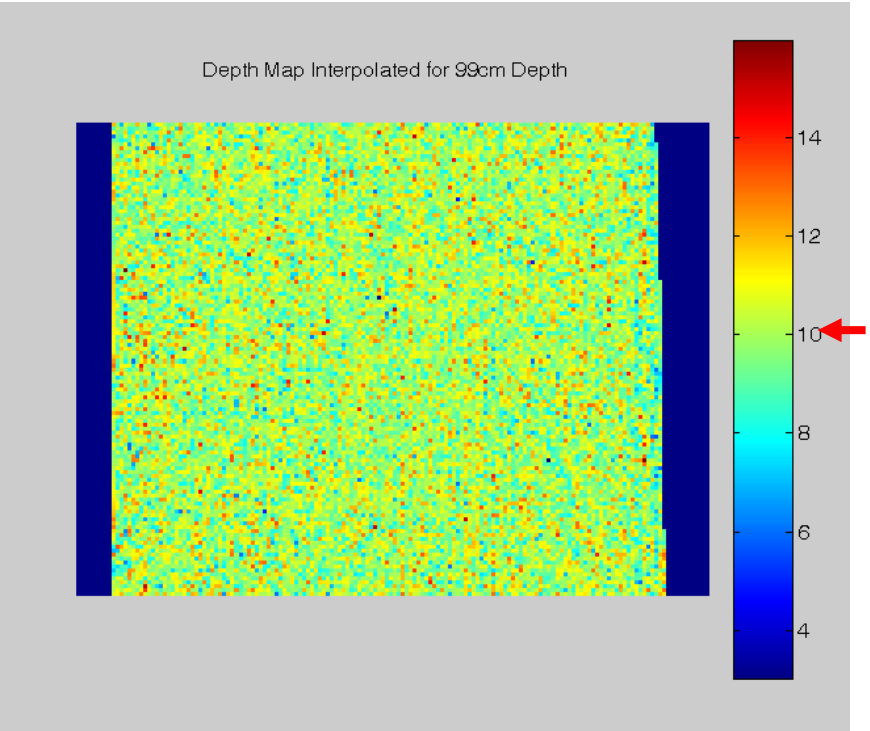
98cm

The known depth is 98 cm
The mean of calculated depth values = 98.0146 cm
The standard deviation of calculated depth values = 1.4992
R.M.S. Error = 1.4992
Percentage R.M.S. Error = 1.5298%
Calculated Depth Estimate Error = 0.014928%



99cm

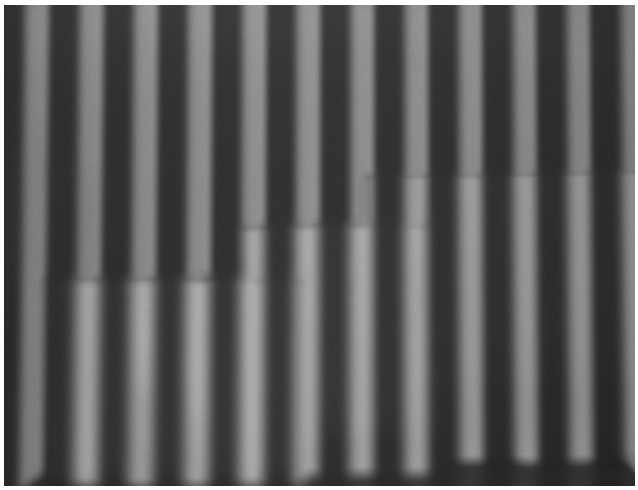
The known depth is 99 cm
The mean of calculated depth values = 99.0382 cm
The standard deviation of calculated depth values = 1.5284
R.M.S. Error = 1.5289
Percentage R.M.S. Error = 1.5443%
Calculated Depth Estimate Error = 0.038569%



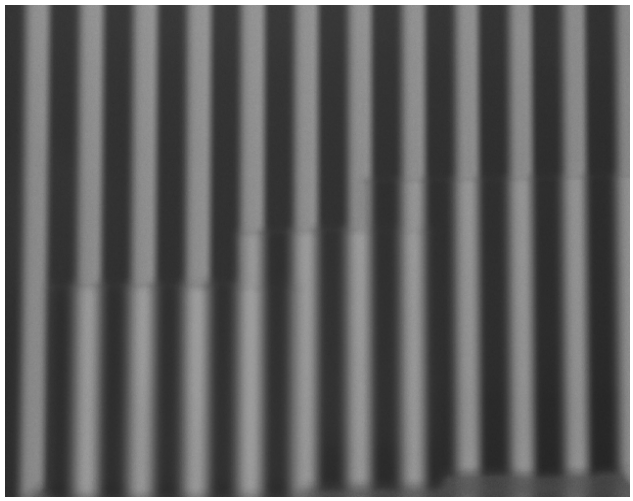
Appendix B-12

The Effect of Texture Neutralisation on an Application Image

(a)



(b)



(c)

Image (a) shows the scene of three planar surfaces at varying depths. The image is illuminated by the data projector set entirely to the intensity value of the 'white bar' of the structured image.

In image (b) the object scene has been illuminated by the structured light pattern, but the texture neutralisation has not yet been applied. The vertical edges of the planar surfaces are masked by the pattern in the position, but the strong horizontal edges are evident.

Image (c) shows the same view as image (b) but after texture neutralisation has been applied. This has normalised the image to the mean of image (a), effectively neutralising the original texture component. Weak horizontal edges of the planar objects are still present as this represents 'additional' texture generated by the illumination of the structured light pattern.

Appendix B-13

Improved Error Statistics for Discrete Look-Up Table Images

Derived from 4-Dimensional Look-up Table using Texture Neutralisation

The values calculated below are derived from the application of the 4-Dimensional DfD system to 17 planar object views at depths 90 cm to 106cm inclusive.
For comparison, the values in brackets are the average error estimates from the previous 2-D LUT.

Image analysed is 090cm08.bmp
The known depth is 90 cm
The mean of calculated depth values = 90.1496 cm
The standard deviation of calculated depth values = 0.859
The ERROR = **0.16627**%
(21-09-06 2-D LUT = 0.69563%)

Image analysed is 091cm08.bmp
The known depth is 91 cm
The mean of calculated depth values = 90.9423 cm
The standard deviation of calculated depth values = 1.0544
The ERROR = **-0.063379**%
(21-09-06 2-D LUT = 0.10335%)

Image analysed is 092cm08.bmp
The known depth is 92 cm
The mean of calculated depth values = 91.9977 cm
The standard deviation of calculated depth values = 1.1152
The ERROR = **-0.0024959**%
(21-09-06 2-D LUT = -0.11581%)

Image analysed is 093cm08.bmp
The known depth is 93 cm
The mean of calculated depth values = 92.9562 cm
The standard deviation of calculated depth values = 1.1403
The ERROR = **-0.04715**%
(21-09-06 2-D LUT = -0.25947%)

Image analysed is 094cm08.bmp
The known depth is 94 cm
The mean of calculated depth values = 93.8914 cm
The standard deviation of calculated depth values = 1.1347
The ERROR = **-0.1155**%
(21-09-06 2-D LUT = -0.071302%)

Image analysed is 095cm08.bmp
The known depth is 95 cm
The mean of calculated depth values = 94.9186 cm
The standard deviation of calculated depth values = 1.1003

The ERROR = **-0.085723**%
(21-09-06 2-D LUT = -0.13308%)

Image analysed is 096cm08.bmp
The known depth is 96 cm
The mean of calculated depth values = 95.8912 cm
The standard deviation of calculated depth values = 1.0709
The ERROR = **-0.11335**%
(21-09-06 2-D LUT = -0.31982%)

Image analysed is 097cm08.bmp
The known depth is 97 cm
The mean of calculated depth values = 96.9906 cm
The standard deviation of calculated depth values = 1.0482
The ERROR = **-0.0096593**%
(21-09-06 2-D LUT = -0.33947%)

Image analysed is 098cm08.bmp
The known depth is 98 cm
The mean of calculated depth values = 98.039 cm
The standard deviation of calculated depth values = 1.0888
The ERROR = **0.039763**%
(21-09-06 2-D LUT = -0.12415%)

Image analysed is 099cm08.bmp
The known depth is 99 cm
The mean of calculated depth values = 98.8766 cm
The standard deviation of calculated depth values = 1.1117
The ERROR = **-0.1246**%
(21-09-06 2-D LUT = -0.48701%)

Image analysed is 100cm08.bmp
The known depth is 100 cm
The mean of calculated depth values = 99.9405 cm
The standard deviation of calculated depth values = 1.0928
The ERROR = **-0.059486**%
(21-09-06 2-D LUT = -0.42714%)

Image analysed is 101cm08.bmp
The known depth is 101 cm
The mean of calculated depth values = 100.9646 cm
The standard deviation of calculated depth values = 1.0961
The ERROR = **-0.035082**%
(21-09-06 2-D LUT = -0.51532%)

Image analysed is 102cm08.bmp
The known depth is 102 cm
The mean of calculated depth values = 102.0349 cm

The standard deviation of calculated depth values = 1.1381
The ERROR = **0.034169**%
(21-09-06 2-D LUT = -0.48751%)

Image analysed is 103cm08.bmp
The known depth is 103 cm
The mean of calculated depth values = 102.938 cm
The standard deviation of calculated depth values = 1.2016
The ERROR = **-0.060197**%
(21-09-06 2-D LUT = -0.74098%)

Image analysed is 104cm08.bmp
The known depth is 104 cm
The mean of calculated depth values = 103.9024 cm
The standard deviation of calculated depth values = 1.2461
The ERROR = **-0.093858**%
(21-09-06 2-D LUT = -0.72436%)

Image analysed is 105cm08.bmp
The known depth is 105 cm
The mean of calculated depth values = 104.918 cm
The standard deviation of calculated depth values = 1.2416
The ERROR = **-0.078069**%
(21-09-06 2-D LUT = -1%)

Image analysed is 106cm08.bmp
The known depth is 106 cm
The mean of calculated depth values = 105.7398 cm
The standard deviation of calculated depth values = 1.0142
The ERROR = **-0.24547**%
(21-09-06 2-D LUT = -1.2172%)

The main purpose of this exercise was to make a comparison of the overall improvement in the performance of the final 4-D system over the original 2-D system, but in order to directly compare the effect of improvement made by the application of texture neutralisation, Figure 5.32 makes a comparison with the results produced in Appendix B-12 (4-D prior to texture neutralisation).
Figure 5.32 shows that there is a general improvement in depth estimation error over the depth range measured, particularly at the smaller depths where the image is more defocused. However, it is noticeable that the Standard Deviation values have improved significantly over the depth range – typically from approximately 1.5 cm to 1.0 cm. Hence, an improvement in the consistency of depth estimate accuracy over the measured area should be achieved.

Appendix C-1

Plus U2 1110 Data Projector

Projection System	DLP
Resolution	1024 x 768 XGA
Brightness	1100 ANSI Lumens
Contrast Ratio	800:1 Full On/Off
Aspect Ratio	4:3
Projection Lens	F=3.0-3.3, f=28-33, 1:1.2 Manual Zoom, Manual Focus
Light Resource	150-Watt High-Performance Compact Lamp **Note:** Projector lamps contain mercury. Do not throw in trash. Dispose of lamps according to local, state, or federal laws.
Color Palette	16.7 Million Colors
Projecting Position	Front/Rear, Table/Ceiling
Projection Distance	3.93 - 40.35', 1.2 - 12.3 m
Image Size	25-300", 635 - 7620 mm
RGB Signal	SXGA Compression, XGA True, SVGA/VGA Expansion/True
Video Signal	NTSC/PAL, NTSC4.43/SECAM
Horizontal Synch Range	15 - 85kHz
Vertical Synch Range	50 - 85 Hz
Audio	Mono 1-Watt
Input Terminals	RGB: HD D-sub 15 pin x 1, Audio (RGB): Stereo Mini Jack x 1, Composite Video: RCA x 1, S-Video: Mini DIN 4 pin x 1, Audio (Video): RCA x 2 (L,R)
Output Terminals	Mouse: Mini DIN 9 pin x 1
Color	Silver
Dimensions	2.28 (H) x 9.25 (W) x 11.7" (L) 58 (H) x 235 (W) x 297mm (L)
Weight	5.6 lbs, 2.5 kg
Power Supply	100-120/220-240 V AC at 50/60 Hz
Power Consumption	220 Watts
On-Screen Display Languages	English, French, German, Italian, Spanish, Swedish, Japanese
Functions	Auto Adjustment, Digital Keystone Correction (15 degrees), Digital Zoom (x 4), Direct Mouse Control, Freeze, Mute, On-Screen Pointers
Accessories	Wireless Remote Control with Laser Pointer, Serial Mouse Cable (3.2'/ 1 m), Mouse Adapter (for IBM PS/2), Mouse Adapter (for Macintosh), Monitor Adapter for Macintosh (DIP-SW), Power Cable (6'/ 1.8 m), RGB Cable (3.2'/ 1 m), Video/Audio Cable (3.2'/ 1 m), S-Video Cable (3.2'/ 1 m), PC Audio Cable (3.2'/ 1 m), Two AA Batteries, Lens Cap, Soft Carrying Case, User's Manual

Appendix C-2

Kyocera – M410R – Digital Camera Specification

Imager	4.0 megapixels effective, 4.23 megapixels gross 1/2.7" square-pixel interlace readout system CCD
Lens	Kyocera 10x zoom lens, 5.7 – 57mm, f/2.8 – 3.1, 11 elements in 9 groups (35mm camera equivalent to approximately 37 – 370mm)
Digital Zoom	1.3x, 1.6x, 2x, 3x, 4x, 5x (1600 x 1200 max res.), 6x (1280 x 960 max res.)
Recording Standards	EXIF 2.2 JPEG, DCF (Design rule for Camera File system)
Shutter	CCD electronic shutter, Programmed electronic shutter with independent aperture control
ISO	Automatic, 100, 200, 400, 800
Exposure Control	Program auto, Aperture priority, Shutter priority: 1/2000 sec. – 1 sec., f/2.8 – f/3.1 Long time: 2, 4 or 8 seconds, f/2.8 – f/3.1
Exposure Correction	+/- 2 EV in 1/3 EV steps
Metering Modes	CCD multi-area evaluation, Center-weighted, Spot
Metering Link Range	LV6 – LV16
Aperture Control	f/2.8 – f/8.0 (at wide-angle lens position)
Focus Control	Video feedback auto focus (Wide/Spot selectable)
Focus Ranges	Wide: 23" – infinity (0.6 m – infinity), Tele: 6.6' – infinity (2.0 m – infinity) Macro wide: 4" – 23" (0.1 m – 0.6 m), Macro tele: 35" – 79" (0.9 m – 2.0 m)
Image Dimensions	Still: 2272 x 1704, 1600 x 1200, 1280 x 960, 640 x 480 Movie: 640 x 480, 320 x 240, 160 x 120
Image Quality	Fine, Normal
White Balance	Automatic, Fixed mode (Preset, Daylight, Overcast, Incandescent, Fluorescent)
Color Modes	Color, Sepia, Black and white
Sharpness	-1 to +3 (5 steps)
Chroma	- / 0 / + (3 steps)
Contrast	- / 0 / + (3 steps)

Appendix C-3

Foculus FO134SB/SC Firewire Digital CCD Camera Specification

Image sensor	1/2" IT CCD ICX414AL	1/2" IT CCD ICX414AQ
Active Pixel	330.000 pixel, 659 (H) x 494 (V) **VGA**	
Data Path	8bit or 12bit B/W	8bit or 12bit Raw RGB/ YUV422
Pixel Size	9.90 (H) x 9.90 (V) µm	
Scanning System	Progressive Scan	
Frame Rate	86 fps (format 7) / 60, 30, 15, 7.5, 3.75, 1.875 fps	
Synchronization	Internal	
Digital Interface	IEEE1394a	
Software Interface	Acc. IIDC v. 1.31	
Gain Control	Manual: 0 ~ 25 dB; Auto Gain	
Gamma	1.0	
Strobe Output Signal	Yes	
S/N Ratio	56 dB or better	
Power Supply/Consumption	+8 to +30 VDC via IEEE1394 Cable	
Transfer rate	400Mbps	
Trigger mode	Software or External Trigger / Mode 0 ~ 5; 14	
Shutter Speed	Manual: 1 µsec ~ 65 sec; Auto shutter	
Advanced Features	Pixel Binning (only B/W); RS232 (SIO/Pass through); ROI; One Shot & Multi Shot; Multi Camera Auto Sync; Opto-Isolated I/O; Industrial Lock Screw Support	
Operating Temperature	- 5°C to +45°C	
Regulations	FCC, CE, RoHS	
Lens Mount	C-Mount	
External Dimensions/Weight	44 (W) x 29 (H) x 63 (D) mm / approx. 110g	
Accessories	• IEEE1394 cable, 6-pin • SDK (Software Development Kit) - free download! • OHCI Card • Tripod Adapter Plate • Trigger Cable 12-pin	

Appendix C-4

IDS uEye UI-1450 Color USB 2.0 Camera Specification

CAMERA SPECIFICATIONS	
	UI-1450-M/C
Resolutions (pixels)	1600x1200
M - Model (b/w)	--
C - Model (colour)	Bayer RGB Pattern
Chip type / Chip size	1/2 Inch/8 mm
Cell size (HxV) in µm	4.2 x 4.2
Shutter	Rolling
Scanning System	Progressive Scan
Sensor dynamics	61 dB
Frames per second	18
Binning	
Subsampling	hor. & vert.
AOI	hor. & vert.
Pixel Clock Frequency (Min/Max)	8/43 Mhz
Shutter speed	45µs - 780ms
Hardware Gain	0-12dB
Hardware Trigger	asynchronous
Lens mount	C-Mount
IR filter	Yes, removable
Interface	USB 2.0
Power supply	< 2 Watts
Operating temperature	0° – 50° Celsius
Regulations	CE, FCC Class B
Size (HxWxD) in mm	34 x 32 x 27,4
Weight	62 g